Hochschule Wismar
Fakultät für Ingenieurwissenschaften
Bereich Elektrotechnik und Informatik

Bachelor-Thesis

Datenschutzkonforme Gestaltung von Open-Source-Kommunikationsplattformen unter Berücksichtigung des Standard-Datenschutzmodells

Eingereicht am: 30. Dezember 2024
von: Stoffi Stoffel
Erstgutachter: Prof. Dr.-Ing. Antje Raab-Düsterhöft
Zweitgutachter: Prof. Dr.-Ing. habil. Andreas Ahrens

Aufgabenstellung

Die Arbeit wird sich mit der Open-Source-Kommunikationsplattform „BigBlue-Button“ unter datenschutzrechtlichen Aspekten beschäftigen. Um die Verarbeitung im Sinne der Datenschutz-Grundverordnung beurteilen zu können, soll durch eine forensische Analyse der Software und ihrer unmittelbaren Komponenten der Umfang der Verarbeitung personenbezogener Daten ermittelt werden. Die daraus gewonnenen Erkenntnisse dienen einer datenschutzrechtlichen Einordnung der Verarbeitung, auf die das Standarddatenschutzmodell (SDM) zur Gewährleistung des Datenschutzes unter technischem Aspekt angewandt werden soll. Die Arbeit wird versuchen, folgende Fragen zu beantworten:

- Welche personenbezogenen Daten werden durch die eingesetzte Software auf welche Weise verarbeitet?
- Wie können die Gewährleistungsziele des Standarddatenschutzmodells umgesetzt werden?
- Welche konkreten technischen Maßnahmen können eingesetzt werden?
- Können die Erkenntnisse in einer generischen Konfigurations- oder Umsetzungshilfe zur Verfügung gestellt werden?

Kurzreferat

Die von der Weltgesundheitsorganisation im März 2020 ausgerufene COVID-19-Pandemie führte weltweit zu einem veränderten Kommunikationsverhalten. Aufgrund der Ansteckungsgefahr verlagerte sich die private und berufliche Kommunikation in den digitalen Raum. Die dabei genutzte kommerzielle bzw. proprietäre Software stieß im Laufe dieser intensiveren Nutzung vermehrt auf datenschutzrechtliche Kritik. Die Datenschutzaufsichtsbehörden empfahlen darum verschiedene Open-Source-Projekte für die digitale Kommunikation. Die Verwendung von freier oder Open-Source-Software muss jedoch ebenfalls kritisch betrachtet und eine mögliche datenschutzkonforme Umsetzung evaluiert werden. Die Aufgabe dieser Arbeit ist die forensische Analyse der Software für eine Beurteilung des Verarbeitungsumfangs personenbezogener Daten im Sinne der Datenschutz-Grundverordnung, die datenschutzrechtliche Einordnung dieser Verarbeitung sowie die Anwendung des Standard-Datenschutzmodells unter dem technischem Aspekt.

Abstract

The COVID-19 pandemic declared by the World Health Organization in March 2020 led to a change in communication behavior worldwide. Due to the risk of infection, private and professional communication shifted to the digital space. The commercial or proprietary software used in this context came under increasing criticism from data protection authorities as use intensified. The data protection supervisory authorities therefore recommended various open source projects for digital communication. However, the use of free or open source software must also be viewed critically and a possible data protection-compliant implementation evaluated. The task of this thesis is the forensic analysis of the software for an assessment of the scope of processing of personal data within the meaning of the General Data Protection Regulation, the classification of this processing under data protection law and the application of the standard data protection model from a technical perspective.

Inhalt

1 Einleitung

Für die digitale Kommunikation steht unterschiedliche Software verschiedenster Anbieter zur Verfügung. Im unternehmerischen und privaten Umfeld wird vielfach auf proprietäre Software US-amerikanischer Unternehmern zurückgegriffen [1,2], weil Teile der angebotenen Softwarepakete bereits im Einsatz sind, sich - auch und gerade im Rahmen von Cloud-Diensten - eine zusätzliche Lizenzierung und Implementierung einfacher gestaltet, ein komfortabler Funktionsumfang geboten wird, die Kommunikation mittels kommerzieller Software vermeintlich professioneller vonstatten geht, Verantwortliche auf bestimmte Wartungs- und Supportangebote zurückgreifen möchten oder die Peergroup einen Nutzungsrahmen setzt. Im Laufe der letzten Jahre gewann Open-Source-Software (OSS) jedoch zunehmend an Bedeutung. [4,5,10] Sie steht im Ruf, durch den offengelegten Quellcode transparent, in ihrer Funktionsweise nachvollziehbar sowie an Nutzungsbedürfnisse anpassbar zu sein. Zudem empfehlen Datenschutzaufsichtsbehörden den Einsatz von OSS-Videokonferenzsystemen. [12,13] Nichtsdestotrotz muss auch diese Software die Anforderungen des Datenschutzes erfüllen. Oftmals sind Dokumentation und Handreichungen für OSS nicht ausreichend oder Datenschutzanforderungen nicht geläufig, so dass auch in diesem Softwareumfeld sog. Datenpannen aufgrund unzureichender Konfigurationen und Maßnahmen nicht vermieden werden können. [7,11]

Die Bedeutung der Arbeit liegt vor allem in der datenschutzrechtlichen Einordnung und der technisch-datenschutzkonformen Umsetzung von OSS im Bereich der Kommunikation mittels Videokonferenzen bzw. Kollaborationsplattformen, mithilfe derer Dokumente gemeinsam bearbeitet, Informationen ausgetauscht, Konferenzen abgehalten und aufgezeichnet werden können. Beim Einsatz solcher Kommunikationsdienste werden serverseitig umfangreiche Daten der beteiligten Personen verarbeitet. Diese können von Nutzungsdaten (auch Daten zur Leistungs- und Verhaltenskontrolle: wann hat welcher Nutzer welche Dienste wie lange in Anspruch genommen) bis zu Inhaltsdaten in Form biometrischer Daten zur Gesichts- und Stimmenerkennung die gesamte Bandbreite der Kategorien personenbezogener Daten umfassen. Verantwortliche benötigen demzufolge für den Einsatz solcher Plattformen eine Einschätzung zum Umfang

der Verarbeitung und zu Möglichkeiten des Schutzes der personenbezogenen Daten.

Neben clientseitigen Artefakten, den Daten, die nur vom Client verarbeitet werden oder darauf verbleiben, wird die Arbeit diese Punkte nicht einbeziehen:

- die datenschutzrechtliche Analyse proprietärer Software,
- eine Quellcodeanalyse der betrachteten Software,
- die Verwendung unterschiedlicher Betriebssysteme sowie Maßnahmen zur Absicherung des jeweiligen Betriebssystems,
- Konfigurationen und Maßnahmen für evtl. beteiligte zentrale Dienste wie Intrusion Detection Systeme, Verzeichnisdienste und ähnliches,
- tiefergehende Analysen zur Absicherung verwendeter Datenbanken und Webserver,
- Vor- und Nachteile sowie Akzeptanz von freier Software und Open-Source-Software,
- KI-basierte Ansätze und Tools
- Anwendungen im gesundheitlichen Bereich.

Es wird kein Web- oder Cloud-Angebot sondern eine eigene Instanz untersucht werden. Zur Umsetzung dieses Vorhabens soll eine Virtuelle Maschine (VM) mit der ausgewählten Software installiert und entsprechend der Empfehlungen des Herstellers konfiguriert werden. Anschließend wird eine Analyse der Prozesse, Dateioperationen, Datenflüsse, der Kommunikation mit den Clients sowie der Daten selbst durchgeführt werden. Die daraus ermittelten Verarbeitungen im Sinne der Datenschutz-Grundverordnung werden in den Maßnahmekatalog des Standarddatenschutzmodells (SDM) eingeordnet. Sofern die eingesetzten Schutzmaßnahmen verbesserungswürdig sind und eine Verbesserung möglich ist, wird versucht werden, dies durch entsprechende Anpassungen herbeizuführen und eine Konfigurationshilfe zur Verfügung zu stellen.

2 Ausgangssituation und Konkretisierung

2.1 Kommunikationsplattformen

Kommunikationsplattformen bieten mehreren Teilnehmern die Möglichkeit, über einen Online-Dienst mithilfe einer App oder eines Browsers virtuell miteinander zu kommunizieren. Unterschieden werden kann hierbei in[1]

- Lernplattformen für den (hoch-)schulischen Bereich, die Lernmaterial, Aufgaben oder Tests zur Verfügung stellen,
- Videokonferenzsysteme, mithilfe derer Online-Konferenzen oder -Meetings durchgeführt werden,
- Messenger-Dienste für den bi- oder multilateralen Nachrichtenaustausch,
- Kurznachrichtendienste (Mikroblogs) für den Austausch von Informationen oder Nachrichten aller Art,
- Foren, die ein allgemeines oder fachspezifisches Spektrum für Fragen und Antworten bieten.

Der Fokus dieser Arbeit liegt auf Lernplattformen und Videokonferenzsystemen. Beide können sich zwar im Funktionsumfang unterscheiden, ermöglichen aber zumeist diese Grundfunktionen: Audio- und Videokommunikation inkl. Aufzeichnung, gemeinsames Bearbeiten von Dokumenten oder Unterhaltung mittels Chat. Das rückt Software in den Mittelpunkt, die zu Beginn der Covid-19-Pandemie einen enormen Nutzungsaufschwung erfuhr. Vielfach wurde der Arbeitsplatz zum „Homeoffice” und geschlossene Schulen führten zur Herausforderung des „Online-Learnings”.

2.2 Motivation zur Konkretisierung

Sowohl im unternehmerischen als auch im (hoch-)schulischen Umfeld wurde zu Beginn der Pandemie vorwiegend auf proprietäre Software US-amerikanischer Anbieter zurückgegriffen. Beispielhaft seien Zoom, Microsoft Teams oder Adobe Connect genannt. Während im kommunalen Bereich die Entwicklung und der

1 vgl. https://serviceportal.eeducation.at/lern-und-kommunikationsplattformen-im-vergleich, https://it.kultus-bw.de/,Lde/Startseite/IT-Sicherheit/Kommunikationsplattformen

Einsatz von datenschutzkonformer Software angestrebt wurde und wird, [14,15,19] wuchs der Nutzungsanteil der proprietären Software in den Unternehmen. [3,9]

Die Datenschutzaufsichtsbehörden kritisierten u. a. rechtliche Vertragsmängel wie Verarbeitung von Daten zu eigenen Zwecken oder nicht rechtmäßige Datenlöschungen, nicht genehmigte Unterauftragsverarbeiter, unzulässige Datenexporte in Drittländer, fehlende Rollenkonzepte oder eine ungenügende Ende-zu-Ende-Verschlüsselung (E2EE). [15,16,18] Wegen der rechtlichen Mängel fand z. T. keine technische Prüfung der Software oder Anbieter statt. Allerdings gab es auch Kritik am Vorgehen der Aufsichtsbehörden, die Zweifel an deren Zuständigkeit für Produktbewertungen hegt, eine ungewisse Sachlichkeit und Richtigkeit in einigen Stellungnahmen vermeint und ein zurückhaltenderes Handeln bei „bei allgemeinen oder technisch nur oberflächlichen Produktbewertungen" begrüßen würde. [22]

Den vielfältigen Funktionsumfang der kritisierten Software kann OSS freilich nicht in Gänze abbilden. Jedoch existieren mit Jitsi Meet[2], Apache OpenMeetings[3], Moodle[4] oder BigBlueButton[5] und anderen durchaus brauchbare Alternativen. Die Datenschutzaufsichtsbehörden begrüßten oder empfahlen mehrfach die Verwendung von BigBlueButton (BBB) und so wurde sich für eine Analyse dieser Software entschieden. BBB ist nach eigener Aussage ein speziell gefertigter virtueller Klassenraum, „der Lehrende zum Lehren und Lernende zum Lernen befähigt".[6] Das entspricht der oben angesprochenen Lernplattform. Jedoch kommt BBB auch in Unternehmen als Kollaborationsplattform oder Videokonferenzsystem zum Einsatz und verlässt damit das rein (hoch-)schulische Umfeld. BBB bietet also einen breiten Anwendungsbereich und zudem ermöglicht die API die Anbindung weiterer Systeme (z. B. Moodle) oder Tools (z. B. WordPress). [21]

2 https://jitsi.org/

3 https://openmeetings.apache.org/

4 https://moodle.org/

5 https://bigbluebutton.org/

6 „Virtual Classroom Software: BigBlueButton is a purpose-built virtual classroom that empowers teachers to teach and learners to learn." so BigBlueButton auf der Webseite, vgl. vorige Fußnote

3 Komponenten und Funktionsweise

3.1 Frontend und Backend

Die Software ermöglicht das Hochladen von Dokumenten, Nutzung von Whiteboards, Gruppenarbeit in speziellen Räumen, Video- und Audiosteuerung, Chats, Umfragen, Teilen von Notizen und Bildschirminhalten und die Nutzung von Webcams inkl. Aufzeichnungen. [17] Diese Funktionen werden durch verschiedene Komponenten realisiert, deren Verarbeitungsergebnis im Browser sicht- und bedienbar sein muss. Die Komponenten sind Teil der Verarbeitung personenbezogener Daten, demzufolge ist ihre Funktionsweise von Interesse, wenn die Verarbeitung personenbezogener Daten eingeordnet werden soll.

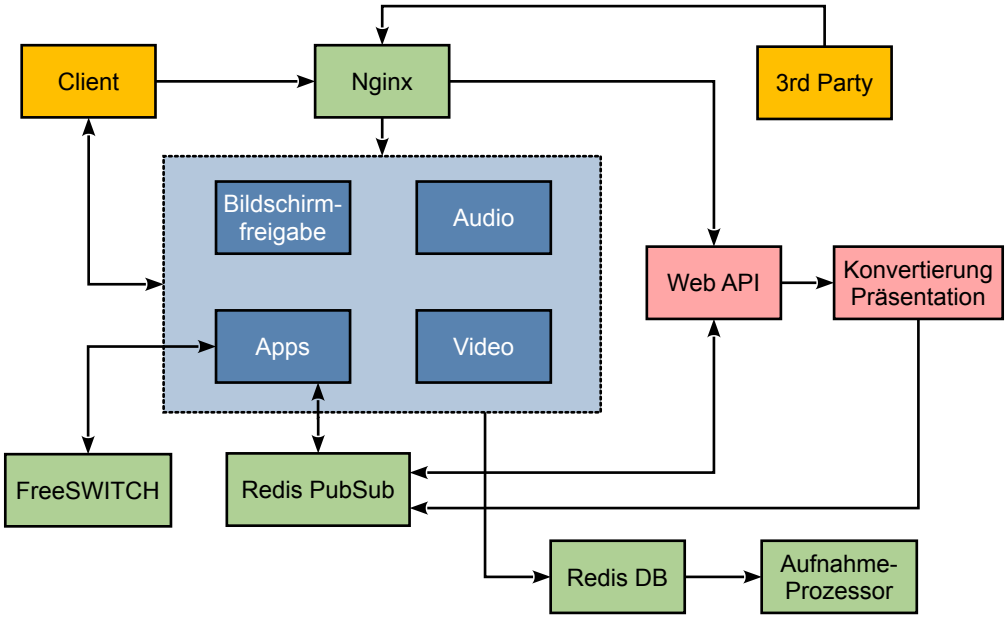

Bild 1 : Allgemeine Architektur der BBB-Komponenten

Laut Beschreibung der Entwickler [20] öffnet ein HTML5-Client über Port 443 (TLS)[7] einen Websocket und verbindet sich direkt mit dem BBB-Server. Ein Websocket ist ein auf TCP basierendes Netzwerkprotokoll, das im Unterschied zu HTTP die Verbindung zwischen Client und Server aufrecht erhält, so dass Daten ohne erneuten Verbindungsaufbau und damit unterbrechungsfrei ausgetauscht werden können. [27] Der Client basiert auf React.js[8] und kommuniziert

7 Die Entwickler schreiben SSL, im Einsatz ist jedoch TLSv1.3, weswegen es hier korrigiert wurde.

8 https://react.dev/

über WebRTC, das die für eine direkte Echtzeitkommunikation notwendigen APIs und Protokolle für „Medienerfassungsgeräte und Peer-to-Peer-Konnektivität" (Videokameras, Mikrofone, Geräte mit Bildschirmaufnahme) [26] zur Verfügung stellt. Die Verbindungen werden vom Webserver nginx[9] verarbeitet.

Die folgende, zusammengefasste Übersicht gibt einen ungefähren Überblick in die Kommunikation zwischen dem HTML5 Client und den Komponenten.

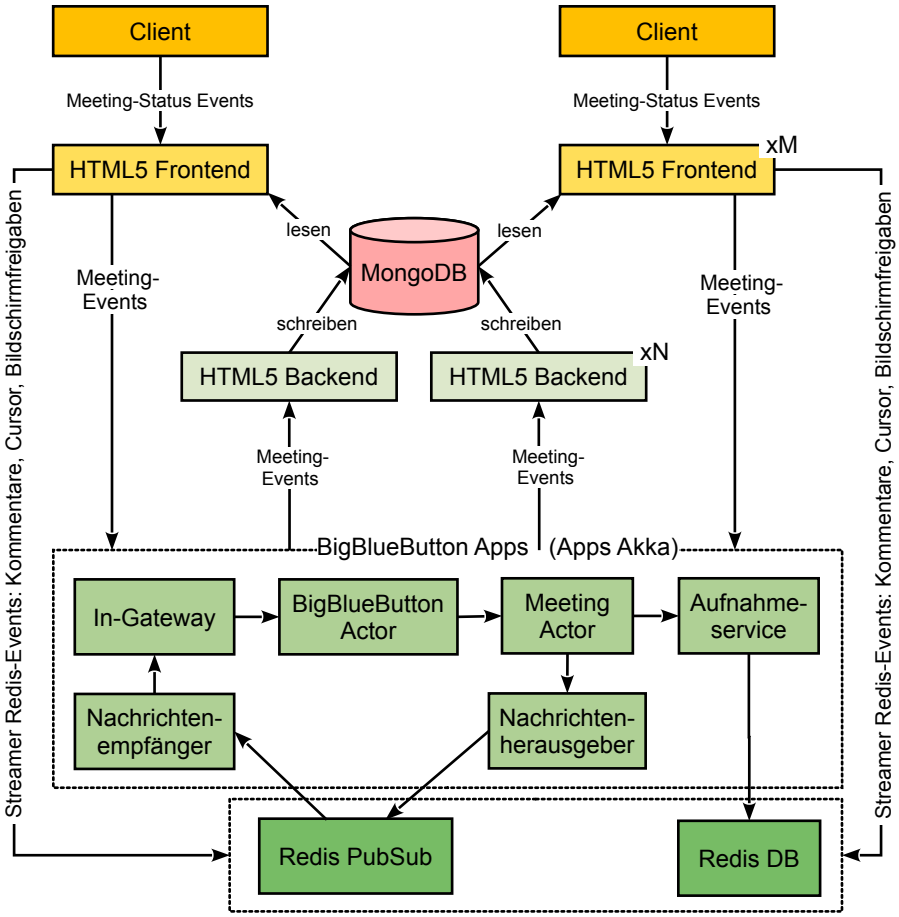

Bild 2 : Kommunikation des HTML5-Clients mit den Komponenten

Im HTML5-Server ist Meteor.js[10] für die Kommunikation zwischen Client und Server implementiert. Für die konsistente Speicherung der Statusinformationen von Meetings (auch Konferenzen), verbundenen Clients und den vom Server bekanntgegebenen Kollektionen ist eine MongoDB zuständig. In BBB-Version 3.0 wird MongoDB nicht mehr eingesetzt werden.[11]

9 https://nginx.org/

10 https://www.meteor.com/

11 vgl. https://docs.bigbluebutton.org/3.0/development/architecture/

FreeSWITCH liefert die Audio-Konferenz-Fähigkeit und ermöglicht die Teilnahme an einem Meeting mit Headset oder per Telefon über das Session Initiation Protocol (SIP). [29] Redis Pub/Sub (publish/subscribe)[12] stellt den Kommunikationskanal zwischen den verschiedenen Applikationen im BBB-Server zur Verfügung. In der Redis-Datenbank werden die Aufzeichnungen in unterschiedlichen RAW-Formaten gespeichert. Im Backend ist außerdem ein Kurento Media Server[13] für die Echtzeitübertragung (streaming) von Webcams, Audiodaten und Bildschirmfreigaben im Einsatz. Er agiert als Medien-Controller und steuert mit seiner MCU- und SFU-Funktionalität (Multipoint Control Units, Selective Forwarding Unit) die verschiedenen Streams der Teilnehmer.

3.2 Interne Netzwerkverbindungen

Ebenfalls interessant für eine Analyse ist die Kommunikation der Komponenten untereinander, weil auch dabei personenbezogene Daten übermittelt werden können. Die nachstehende Übersicht zeigt die Verbindungen der verschieden Komponenten von BBB.

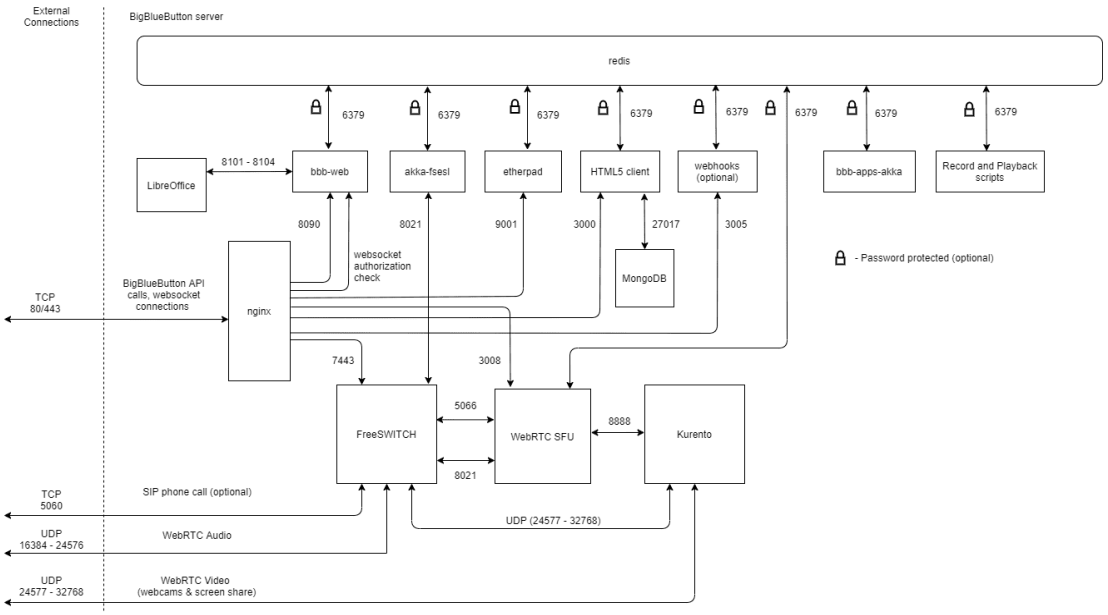

Bild 3 : Kommunikation der BBB-Komponenten, Quelle: BigBlueButton

12 https://redis.io/docs/latest/develop/interact/pubsub/

13 https://github.com/Kurento/kurento

4 Forensische Analyse der Verarbeitung personenbezogener Daten

Eine exakte Widerspiegelung aller verwendeten Methoden und Befehle wäre für diese Arbeit zu umfangreich. Aus diesem Grund wird die Vorgehensweise lediglich exemplarisch beschrieben, so dass sie nachvollzogen werden kann. Zunächst wurde die Grundlage aller Verarbeitungen erfasst: die zu BBB gehörenden Prozesse. Daraus ließen sich Dateioperationen ableiten, konnten die Kommunikation sowie der Inhalt von Datenbanken analysiert und schließlich Rückschlüsse zu Verarbeitung und Verarbeitungsprozessen gezogen werden. Um sich einen möglichst unabhängigen Überblick zu verschaffen, wurde die Dokumentation des Herstellers nur noch zu Erklärungszwecken herangezogen.

4.1 Laborumgebung

Für die Analyse wurde entsprechend der Empfehlungen[14] von BigBlueButton eine VM namens „blue" in einem eigens dafür eingerichteten Subnetz mit der Domain „tuxbook.de" aufgesetzt und darin die BBB-Version 2.7 installiert. Zur Überwachung der Kommunikation mit den Clients kam eine VM mit Kali GNU/Linux (Kali-VM) zum Einsatz. Sie sollte als Man-In-The-Middle (MITM) die verschlüsselte Kommunikation mit den Clients lesbar machen. Im Folgenden wird die Herangehensweise exemplarisch beschrieben.

4.1.1 Installation BigBlueButton

Die Installation von BBB fand gemäß der Installationsanleitung statt. Hierfür wurde das Script ‚bbb-install.sh' aus dem Github-Repository heruntergeladen und als root mit folgenden Optionen ausgeführt:

```
# ./bbb-install.sh -s -- -v focal-270 -s blue.tuxbook.de -e user@tuxbook.de -w -g
  -d
```

Auf die Installation von Keycloak[15] zur Einbindung von erweiterten Authentifizierungsmethoden wurde verzichtet. Da dies keine zwingende Voraussetzung für

14 BigBlueButton, Installation, Minimum server requirements
https://docs.bigbluebutton.org/administration/install/#minimum-server-requirements

15 https://www.keycloak.org/

die Funktionalität von BBB ist und eine Analyse der Verarbeitung personenbezogener Daten in Keycloak, den Protokollen OIDC[16], OAuth[17] oder SAML[18] sowie der verschiedensten Authentifizierungsanbieter wie LDAP, Active Directory oder Social Logins zu umfangreich wäre. Für das Vorhaben genügten lokal angelegte Nutzer. Dafür wurden in der BBB-VM vier fiktive Nutzer angelegt: black, green, lila und red. Im jeweiligen Home-Verzeichnis der Nutzer befand sich auch das E-Mail-Postfach, so dass die von BBB an die Nutzer gesendeten E-Mails eingesehen werden konnten.

Die VM wurde für den BBB-Betrieb angepasst:

- Es wurde IPv6 abgeschaltet.
- Ein lokaler Mailserver und die Authentisierung per SASL wurde eingerichtet; der Greenlight-Container und die Firewall für den E-Mail-Versand konfiguriert. Das Simple Authentication and Security Layer (SASL) ist ein Framework zur Bereitstellung von Authentifizierungs- und Datensicherheitsdiensten in verbindungsorientierten Protokollen über austauschbare Mechanismen [8].
- Für die Nutzung im lokalen Subnetz wurden eigene Zertifikate generiert, den Komponenten bekanntgemacht und das Image des Greenlight-Containers neu erstellt.

Eine detaillierte Übersicht der Änderungen kann der Anlage Anpassung der VM für den BigBlueButton-Betrieb entnommen werden.

4.1.2 Einrichtung Kali GNU/Linux

Die Kommunikation der Clients mit BBB ist deshalb von Interesse, weil hierbei personenbezogene Daten, z. B. Logindaten, Session-Token oder Cookies, übermittelt werden. Diese Informationen können größtenteils mit den Entwickler-Tools des Browsers erfasst werden, jedoch ist es bei der Nutzung mehrerer

16 https://openid.net/developers/how-connect-works/

17 https://oauth.net/2/

18 https://www.rfc-editor.org/rfc/rfc7522.html

Clients effizienter, einen zentralen Kommunikationspunkt für den Mitschnitt und dessen Speicherung zu verwenden. Um einen Einblick in die Datenübertragung erhalten zu können, wurde die Version 2024.3 von Kali GNU/Linux in einer VM installiert (Kali-VM).

Grundsätzlich bieten die Tools der Kali-VM mehrere Möglichkeiten, die Kommunikation von Beteiligten mitzulesen. Mit arpspoof[19] kann z. B. der Standardgateway ersetzt und die Kommunikation über einen Dritten geleitet werden, der die Übertragung mit sslsplit[20] entschlüsseln und mitlesen könnte. Es wurde sich jedoch für die Nutzung von mitmproxy[21] entschieden, da diese Methode mit den geringsten Eingriffen bei den beteiligten VMs und Nutzern verbunden war. Die auf der Kali-VM vorhandenen Python-Scripte für mitmproxy funktionierten auch nach einigen Anpassungen nicht, weswegen die Original-Binaries von der Webseite mitmproxy.org heruntergeladen und genutzt wurden.

Nach dem erstem Start von mitmproxy wird im Home-Verzeichnis des Nutzers „kali" ein Unterverzeichnis „.mitmproxy" mit Zertifikaten erzeugt. Das darin befindliche CA-Zertifikat „mitmproxy-ca-cert.pem" fand im späteren Verlauf Verwendung. Zur Umgehung des nicht vertrauenswürdigen, selbsterstellten Zertifikats wurde mitmproxy mit der Option „--ssl-insecure" gestartet.

```
# sudo sysctl -w net.ipv4.ip_forward=1
# sudo iptables -t nat -F
# sudo iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j REDIRECT --
  to-port 8080
# sudo iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --
  to-port 8080
# mitmweb --ssl-insecure
```

Da es nicht auf eine unbemerkbare Kompromittierung ankam, sondern das Augenmerk allein auf dem Inhalt der Übermittlung lag, wurde die einfachste Methode zum Mitlesen gewählt: Bei allen Browsern wurde die Kali-VM als Proxy-Server eingetragen und das CA-Zertifikat „mitmproxy-ca-cert.pem" importiert, so dass diese den Proxy ohne Fehlermeldung akzeptierten und die Kommunikation überwacht werden konnte.

19 https://github.com/smikims/arpspoof

20 https://github.com/droe/sslsplit

21 https://mitmproxy.org/

4.2 Erfassung der Prozesse

Aus der Serviceabfrage mittels ‚systemctl' wurden folgende Services im Status „loaded active running" erfasst:[22]

```
# systemctl --type service --state running|egrep -v <UBUNTU-SERVICES>
```

```
UNIT                              LOAD   ACTIVE SUB     DESCRIPTION
bbb-apps-akka.service             loaded active running BigBlueButton Apps (Akka)
bbb-export-annotations.service    loaded active running BigBlueButton Export Annotations
bbb-fsesl-akka.service            loaded active running BigBlueButton FS-ESL (Akka)
bbb-html5-backend@1.service       loaded active running BigBlueButton HTML5 service, backend instance 1
bbb-html5-backend@2.service       loaded active running BigBlueButton HTML5 service, backend instance 2
bbb-html5-frontend@1.service      loaded active running BigBlueButton HTML5 service, frontend instance 1
bbb-html5-frontend@2.service      loaded active running BigBlueButton HTML5 service, frontend instance 2
bbb-pads.service                  loaded active running BigBlueButton Pads
bbb-rap-caption-inbox.service     loaded active running BigBlueButton recording caption upload handler
bbb-rap-resque-worker.service     loaded active running BigBlueButton resque worker for recordings
bbb-rap-starter.service           loaded active running BigBlueButton recording processing starter
bbb-web.service                   loaded active running BigBlueButton Web Application
bbb-webrtc-recorder.service       loaded active running BigBlueButton WebRTC Recorder
bbb-webrtc-sfu.service            loaded active running BigBlueButton WebRTC SFU
```

Bild 4 : Screenshot der laufenden systemd-Services

In der Prozessliste waren neben den aus den Servicenamen erwartbaren Prozesse außerdem Ruby-, Java- und Node-Prozesse zu sehen:

```
    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
228699 root       20   0 10260  5712  3456 R  2.6  0.1  0:13.07 htop
   1003 bigbluebu 20   0 4153M  748M 24664 S  1.9  9.4  1h32:24 /usr/bin/java -Dgrails.env=prod -Dserver.addre
    967 freeswitc -2  19 1665M 40644 14664 S  1.9  0.5  1h24:42 /opt/freeswitch/bin/freeswitch -u freeswitch -
   1376 bigbluebu 20   0 3879M  350M 23848 S  1.3  4.4 21h52:08 java -Xms130m -Xmx256m -Dconfig.file=/etc/bigb
   1365 bigbluebu 20   0 3887M  280M 24000 S  1.3  3.5  1h23:23 java -Xms130m -Xmx256m -Dconfig.file=/etc/bigb
    913 mongodb    20   0 1001M  142M 44652 S  1.3  1.8  1h47:30 /usr/bin/mongod --config /usr/share/meteor/bun
   3231 bigbluebu 20   0 3887M  280M 24000 S  1.3  3.5 56:20.80 java -Xms130m -Xmx256m -Dconfig.file=/etc/bigb
   9379 bigbluebu 20   0 4153M  748M 24664 S  1.3  9.4 52:19.42 /usr/bin/java -Dgrails.env=prod -Dserver.addre
   3115 bigbluebu 20   0 3879M  350M 23848 S  0.6  4.4 53:27.22 java -Xms130m -Xmx256m -Dconfig.file=/etc/bigb
    944 redis      20   0 53108  5492  3600 S  0.6  0.1 25:42.57 /usr/bin/redis-server 127.0.0.1:6379
218787 freeswitc -2  19 1665M 40644 14664 S  0.6  0.5  0:03.08 /opt/freeswitch/bin/freeswitch -u freeswitch -
   1936 mongodb    20   0 1001M  142M 44652 S  0.6  1.8 15:17.35 /usr/bin/mongod --config /usr/share/meteor/bun
197526 mongodb    20   0 1001M  142M 44652 S  0.6  1.8  0:08.82 /usr/bin/mongod --config /usr/share/meteor/bun
   9828 bigbluebu 20   0 4153M  748M 24664 S  0.6  9.4  4:58.26 /usr/bin/java -Dgrails.env=prod -Dserver.addre
   1083 freeswitc -2  19 1665M 40644 14664 S  0.6  0.5  6:03.83 /opt/freeswitch/bin/freeswitch -u freeswitch -
   1866 mongodb    20   0 1001M  142M 44652 S  0.6  1.8  3:51.31 /usr/bin/mongod --config /usr/share/meteor/bun
218732 bigbluebu -2  19 10.9G 67660 39980 S  0.6  0.8  0:01.54 /usr/bin/node ./lib/screenshare/ScreenshareProv
175686 bigbluebu 10 -10 1430M 14812 11248 S  0.6  0.2  0:27.19 /usr/bin/bbb-webrtc-recorder
175702 bigbluebu 10 -10 1430M 14812 11248 S  0.6  0.2  0:03.38 /usr/bin/bbb-webrtc-recorder
  4825 freeswitc -2  19 1665M 40644 14664 S  0.0  0.5 21:58.55 /opt/freeswitch/bin/freeswitch -u freeswitch -
  2670 systemd-c 20   0 30408  5964  4484 S  0.0  0.1 17:08.79 redis-server *:6379
  2696 meteor    -2  18  755M  121M 32856 S  0.0  1.5 15:21.19 /usr/lib/bbb-html5/node/bin/node --max-old-spa
  1352 freeswitc RT  19 1665M 40644 14664 S  0.0  0.5 18:32.91 /opt/freeswitch/bin/freeswitch -u freeswitch -
  2691 meteor    -2  18  819M  122M 32776 S  0.0  1.5 15:10.94 /usr/lib/bbb-html5/node/bin/node --max-old-spa
  9900 bigbluebu 20   0 4153M  748M 24664 S  0.0  9.4  4:59.02 /usr/bin/java -Dgrails.env=prod -Dserver.addre
  4260 bigbluebu 20   0 3887M  280M 24000 S  0.0  3.5  5:05.60 java -Xms130m -Xmx256m -Dconfig.file=/etc/bigb
  1254 mongodb    20   0 1001M  142M 44652 S  0.0  1.8  5:41.40 /usr/bin/mongod --config /usr/share/meteor/bun
  2620 root       20   0  223M  120M 11916 S  0.0  1.5  1:40.46 puma 5.6.8 (tcp://0.0.0.0:3000) [app]
```

Bild 5 : Screenshot Prozessansicht von htop sortiert nach CPU

Mit den Befehlen ‚ps' (report a snapshot of the current processes) und ‚lsof' (list open files) wurden die zu BBB gehörenden Prozesse ermittelt. Beide Befehle sind Linux-Standards, wobei ‚lsof' eine tiefergehende Analyse erlaubt, da unter Linux alles - ob Geräte, Verzeichnisse, Dateien, Sockets oder benannte Pipes -

22 <UBUNTU-SERVICES> beinhaltet nicht zu BBB gehörende Services.

als Datei repräsentiert wird. [40] Dadurch können mit einer Auflistung offener Dateien umfangreiche Informationen zu einem Prozess gewonnen werden. Exemplarisch wird die Ermittlung der Prozesse anhand eines im Docker-Container und eines auf dem Host laufenden Prozesses beschrieben.

4.2.1 Container-Prozesse

Aus der Service-Liste konnte die Existenz eines Docker-Service' entnommen werden. Eine Container-Abfrage listete diese Instanzen auf:[23]

```
# docker ps
CONTAINER ID    IMAGE                                NAMES
83abd5def980    bigbluebutton/greenlight:v3          greenlight-v3
fae6b7e08263    redis:6.2-alpine3.17                 redis
b417bcf9adad    postgres:14.6-alpine3.17             postgres
```

Docker-Prozesse können auf dem Host mit dem Befehl ‚systemd-cgls' oder mit den Docker-eigenen Befehlen ermittelt werden:

```
# systemd-cgls
├─docker
│ ├─b417bcf9adad82b14abdd88d736f94370c918b2b52b1e8fbd224fe3c9b199d84
│ │ ├─2592 postgres
...
│ │ ├─2949 postgres: walwriter
│ │ ├─2950 postgres: autovacuum launcher
│ │ ├─2951 postgres: stats collector
│ │ └─2952 postgres: logical replication launcher
│ ├─fae6b7e082630e80fe3d3b387a4966038b3d139dd20ea3fbb7ad575f4632a2fd
│ │ └─2670 redis-server *:6379
│ └─83abd5def980ac208e539db7152f8e653582688c78586ab0e41ef783637feb8a
│   └─2620 puma 5.6.8 (tcp://0.0.0.0:3000) [app]
```

Der Vorteil bei der Verwendung Docker-eigener Befehle besteht darin, dass explizite Container-Informationen abgefragt und für automatisierte Funktionen weiterverwendet werden können.

```
# for i in $(docker container ls --format "{{.ID}}"); do
  echo -e "\nID: $i -------"
  docker inspect -f '{{.State.Pid}}' $i|xargs pstree -psa
done

ID: 83abd5def980 -------
systemd,1 maybe-ubiquity
  └─containerd-shim,2515 -namespace moby -id
  83abd5def980ac208e539db7152f8e653582688c78586ab0e41ef783637feb8a -address
  /run/containerd/containerd.sock
      └─ruby,2620
          ├─{ruby},7237
          ├─{ruby},7413
...
```

23 Auf die Ausgabe von COMMAND, CREATED, PORT und STATUS wurde aus Platzgründen verzichtet.

```
ID: fae6b7e08263 -------
systemd,1 maybe-ubiquity
  └─containerd-shim,2493 -namespace moby -id
  fae6b7e082630e80fe3d3b387a4966038b3d139dd20ea3fbb7ad575f4632a2fd -address
  /run/containerd/containerd.sock
       └─redis-server,2670
           ├─{redis-server},2902
           ├─{redis-server},2903
...

ID: b417bcf9adad -------
systemd,1 maybe-ubiquity
  └─containerd-shim,2475 -namespace moby -id
  b417bcf9adad82b14abdd88d736f94370c918b2b52b1e8fbd224fe3c9b199d84 -address
  /run/containerd/containerd.sock
       └─postgres,2592
           ├─postgres,2946
           ├─postgres,2947
...
```

Der Container mit ID „83abd5def980" enthält die Greenlight-Prozesse, welche auf Ruby on Rails[24] und React.js basieren. [38] Greenlight ist eine Open-Source Webapplikation zum Einrichten einer Webkonferenz-Plattform für BBB. Die obige Ausgabe der Docker-Prozesse listet den Ruby-Prozess mit Prozess-ID (PID) 2620 (hervorgehoben). Die folgende Ausgabe zeigt neben der PID des Prozesses auch die PID des Elternprozesses (2515) und das ausgeführte Kommando.

```
Host: # ps -q 2620 -o pid,ppid,cmd
    PID    PPID CMD
   2620    2515 puma 5.6.8 (tcp://0.0.0.0:3000) [app]
```

```
root@blue:~# lsof -p 2620
COMMAND  PID USER   FD      TYPE DEVICE SIZE/OFF    NODE NAME
ruby    2620 root  cwd       DIR   0,59     4096 2623678 /usr/src/app
ruby    2620 root  rtd       DIR   0,59     4096 2624886 /
ruby    2620 root  txt       REG   0,59   160128 2492632 /usr/local/bin/ruby
ruby    2620 root  mem       REG   0,59          2501620 /usr/src/app/vendor/bundle/ru
ruby    2620 root  mem       REG   0,59          2504701 /usr/src/app/vendor/bundle/ru
ruby    2620 root  mem       REG   0,59          2494075 /usr/local/lib/ruby/3.2.0/x86
ruby    2620 root  mem       REG   0,59          2500640 /usr/src/app/vendor/bundle/ru
ruby    2620 root  mem       REG   0,59          2502519 /usr/src/app/vendor/bundle/ru
ruby    2620 root    0u      CHR    1,3       0t0       6 /dev/null
ruby    2620 root    1w     FIFO   0,13       0t0   37018 pipe
ruby    2620 root    2w     FIFO   0,13       0t0   37019 pipe
ruby    2620 root    3u  a_inode   0,14         0   10302 [eventfd]
ruby    2620 root    4u  a_inode   0,14         0   10302 [eventfd]
ruby    2620 root    5u     sock    0,9       0t0   44069 protocol: TCP
ruby    2620 root    6r     FIFO   0,13       0t0   44070 pipe
ruby    2620 root    7w     FIFO   0,13       0t0   44070 pipe
ruby    2620 root    8u  a_inode   0,14         0   10302 [eventpoll]
ruby    2620 root    9r     FIFO   0,13       0t0   44071 pipe
ruby    2620 root   10w     FIFO   0,13       0t0   44071 pipe
ruby    2620 root   20u     sock    0,9       0t0 1319477 protocol: TCP
```

Bild 6 : Screenshot Host lsof-Ausgabe für den Ruby-Prozess von Greenlight (gekürzt)

Bild 6 veranschaulicht die Ausgabe von ‚lsof' für PID 2620 des Hosts. Im Container ist die PID des Ruby-Prozesses eine andere, nämlich die 1. PID 1 ist un-

24 https://rubyonrails.org/

ter Linux der init-Prozess und der erste User-Mode-Prozess, der gestartet wird und bis zum Ausschalten des Systems läuft. [39]:

```
Container: # ps -ef -o pid,ppid,cmd
    PID    PPID CMD
      1       0 puma 5.6.8 (tcp://0.0.0.0:3000) [app]
```

Auch die Ausgabe von ‚lsof' unterscheidet sich (siehe Bild 7). Während auf dem Host die in den Arbeitsspeicher geladenen Ruby-Bibliotheken, FIFOs, Sockets und symbolische Links zu Dateideskriptoren angezeigt werden, enthält die Ausgabe im Container keine Bibliotheken; die FIFO- und Socket-Inodes stimmen überein.

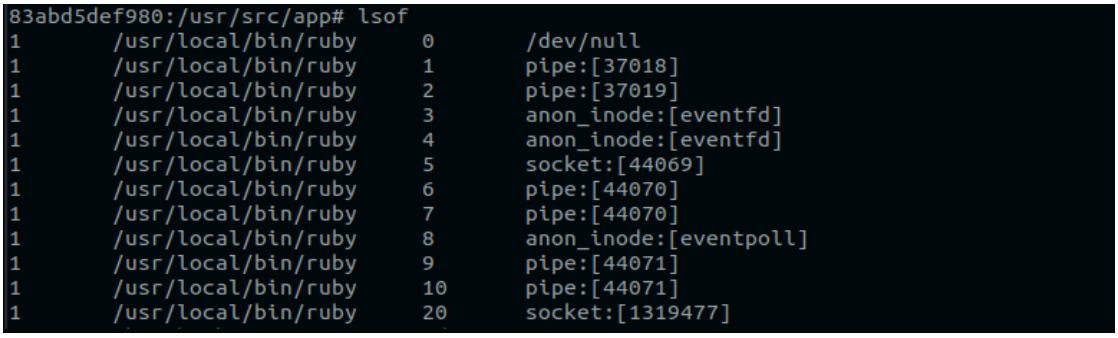

Bild 7 : Screenshot Container lsof-Ausgabe für den Ruby-Prozess von Greenlight (gekürzt)

4.2.2 Host-Prozesse

Eine weitere Möglichkeit, Informationen zu Prozessen zu erhalten, ist die Analyse der unter /proc vorhandenen Daten. Die Host-Prozesse wurden jedoch auf dieselbe Weise wie die Container-Prozesse erfasst. Die dem Nutzer root gehörenden Prozesse von BBB waren Docker-, haproxy- und nginx-Prozesse. Mit dieser Information konnten die Nutzer der Prozesse ermittelt werden.

```
# ps -axwwo user:32|awk '{print $1}'|sort -n|uniq|egrep -v <UBUNTU-USERS>[25]
bigbluebutton
etherpad
freeswitch
haproxy
kurento
meteor
mongodb
redis
syslog
turnserver
www-data
70
```

25 <UBUNTU-USERS> beinhaltet nicht zu BBB gehörende Nutzer.

Nutzer „70” ist der Nutzer „postgres” des PostgreSQL-Containers, der dem Host nicht bekannt ist, darum wird die UID statt des Nutzernamens angezeigt. Anhand der obigen Nutzerliste war eine gezielte Auswahl von Prozessen möglich. Hier ein Beispiel für den Nutzer „bigbluebutton”:

```
root@blue:~# ps -u bigbluebutton -o pid,ppid,cmd
    PID    PPID CMD
    776       1 /usr/bin/node master.js
    777       1 /usr/bin/ruby /usr/local/bigbluebutton/core/scripts/rap-caption-inbo
    994       1 ruby2.7 /usr/local/bigbluebutton/core/vendor/bundle/ruby/2.7.0/bin/r
    998       1 /usr/bin/ruby /usr/local/bigbluebutton/core/scripts/rap-starter.rb
   1003       1 /usr/bin/java -Dgrails.env=prod -Dserver.address=127.0.0.1 -Dserver.
   1005       1 /usr/bin/node index.js
   1365       1 java -Xms130m -Xmx256m -Dconfig.file=/etc/bigbluebutton/bbb-fsesl-ak
   1376       1 java -Xms130m -Xmx256m -Dconfig.file=/etc/bigbluebutton/bbb-apps-akk
   1786     994 resque-2.6.0: Waiting for rap:archive,rap:publish,rap:process,rap:sa
 236884       1 /usr/bin/bbb-webrtc-recorder
 239115       1 /usr/bin/node server.js
 239144  239115 /usr/bin/node ./lib/mcs-core/process.js
```

Bild 8 : Screenshot ps-Ausgabe Host-Prozesse von Nutzer „bigbluebutton” (gekürzt)

Für jeden einzelnen dieser Prozesse konnten mit ‚lsof‘ weitere Informationen gewonnen werden, die für den späteren Verlauf der Analyse relevant waren (z. B. TCP-Verbindungen, Log-Dateien).

Das Beispiel in Bild 9 zeigt die Informationen für den Screenshare-Prozess des Nutzers „bigbluebutton”.

```
PID     FD      TYPE    NODE            NAME
239145  mem     REG     12588734        /usr/lib/x86_64-linux-gnu/ld-2.31.so
239145  mem     REG     12592059        /usr/lib/x86_64-linux-gnu/libgcc_s.so.1
239145  mem     REG     12592060        /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.
239145  mem     REG     12592065        /usr/lib/x86_64-linux-gnu/libc-2.31.so
239145  21w     FIFO    1389584         pipe
239145  1u      unix    1390628         type=STREAM
239145  2u      unix    1390628         type=STREAM
239145  3u      unix    1390646         type=STREAM
239145  28w     REG     1572873         /var/log/bbb-webrtc-sfu/bbb-webrtc-sfu.log
239145  27u     IPv4    TCP             localhost:3022
239145  23u     IPv4    TCP             localhost:50688->localhost:6379
239145  24u     IPv4    TCP             localhost:50694->localhost:6379
239145  25u     IPv4    TCP             localhost:50706->localhost:6379
```

Bild 9 : Screenshot lsof-Ausgabe Screenshare-Prozess von „bigbluebutton” (gekürzt)

4.3 Erfassung der internen Kommunikation

Die aus 4.2 gewonnen Informationen der beteiligten Prozesse ermöglichten eine Zuordnung von Prozessen und ihrer Kommunikation (siehe Bild 9 IPv4 TCP). Mit den Informationen aus /proc/net/ oder den Befehlen ‚netstat‘ und ‚ss‘

kann die interne Kommunikation ebenfalls ermittelt werden. Zur Erfassung aller TCP- und UDP-Ports im Status „LISTEN" (lauschend, hörend) ist ‚netstat' ausreichend.

```
root@blue:~# netstat -tulpen
Active Internet connections (or
Proto      Local Address          State       User       Inode      PID/Program name
tcp        127.0.0.1:4101         LISTEN      995        38313      2691/node
tcp        127.0.0.1:3014         LISTEN      996        1389651    239144/node
tcp        127.0.0.1:3016         LISTEN      996        1390660    239115/node
tcp        127.0.0.1:9001         LISTEN      997        42166      1004/node
tcp        127.0.1.1:27017        LISTEN      114        30170      913/mongod
tcp        127.0.0.1:9002         LISTEN      996        41396      1005/node
```

Bild 10 : Screenshot netstat-Ausgabe für Ports im Status LISTEN (gekürzt)

Die damit ermittelten Ports sollten mit denen in Bild 3 übereinstimmen. Das war allerdings nicht der Fall, was evtl. auf eine veraltete Dokumentation zurückzuführen ist. Je nach Vorgehensweise können genutzte Verbindungen auch mit ‚lsof' angezeigt werden. Hier ein Beispiel für offene UDP-Verbindungen:

```
# lsof -i UDP
COMMAND     PID  USER        FD   TYPE DEVICE SIZE/OFF NODE NAME
...
turnserve   805  turnserver  28u  IPv4  44033      0t0  UDP blue:3478
turnserve   805  turnserver  29u  IPv4  44034      0t0  UDP blue:3478
turnserve   805  turnserver  30u  IPv4  44035      0t0  UDP blue:3478
turnserve   805  turnserver  31u  IPv4  44036      0t0  UDP blue:3478
freeswitc 36106  freeswitch  17u  IPv4 356829      0t0  UDP blue:5090
freeswitc 36106  freeswitch  22u  IPv4 354537      0t0  UDP blue:sip
...
```

Anhand der mit ‚netstat' oder ‚lsof' ermittelten Ports wurde mit ‚tcpdump' die Kommunikation der einzelnen Komponenten während verschiedener Meetingaktivitäten mitgeschnitten und in Wireshark analysiert. Bild 11 zeigt den Mitschnitt des von Redis genutzten Ports 6379.

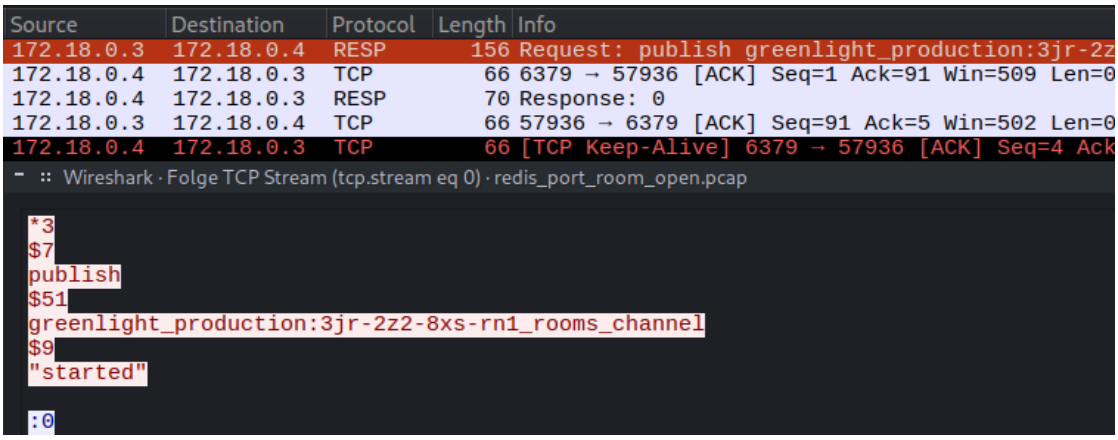

Bild 11 : Screenshot Wireshark: Auszug eines tcpdump-Mitschnitts Redis-Port 6379

Hier wurden die Eigenschaften eines Raumes zwischen Greenlight und Redis-Server übertragen. Das Protokoll „RESP” (Redis Serialization Protocol) ist ein für Redis entwickeltes Protokoll, das technisch gesehen nicht TCP-spezifisch ist, aber im Redis-Kontext exklusiv mit TCP-Verbindungen genutzt wird. [51]

4.4 Erfassung von Dateioperationen

Durch die Prozesserfassung mit ‚lsof‘ und die Nutzung der Informationen zu Dateideskriptoren unter /proc war auch eine Ermittlung genutzter Dateien möglich. Hierbei kam es auf die Unterscheidung zwischen regulären Dateien, Sockets und Pipes an, denn es war beabsichtigt, die regulären Dateien eines Prozesses ausfindig zu machen und als Artefakt für weitere Analysen zu registrieren. Dem ‚lsof‘-Befehl können zu diesem Zweck Dateideskriptoren übergeben werden. Das sind vom Kernel vergebene eindeutige, positive ganze Zahlen, die Dateien zum Zweck des Zugriffs repräsentieren. [41] Das folgende Beispiel mit ‚lsof‘ zeigt eine Liste offener regulärer Dateien für die PID 967. Als Parameter kann dem Befehl ein Zahlenbereich zwischen dem kleinsten und größten Dateideskriptor übergeben werden. „Standard Input”, „Standard Output” und „Standard Error” belegen immer 0 - 2, weswegen diese Zahlen für die Suche nach regulären Dateien irrelevant sind. Für die Ermittlung des höchsten Dateideskriptors eines Prozesses können die Informationen aus /proc genutzt und an ‚lsof‘ übergeben werden. Die Einschränkung mit ‚grep‘ stellt sicher, dass nur Dateien mit schreibenden Zugriff aufgelistet werden[26].

```
# lsof -a -d 3-$(ls /proc/967/fd|sort -n|tail -1) -p 967 /|grep -E '[0-9](u|w)'
FD       TYPE   NAME
5u       REG    /opt/freeswitch/var/log/freeswitch/freeswitch.log
13u      REG    /opt/freeswitch/var/lib/freeswitch/db/sofia_reg_internal.db
26u      REG    /opt/freeswitch/var/lib/freeswitch/db/sofia_reg_internal.db
```

Dieselbe Information enthält die Auflistung der Dateideskriptoren unter /proc:

```
# ll /proc/967/fd|grep -E "/[a-z]"|egrep -v 'dev|proc|pid'|awk '{print $NF}'
/opt/freeswitch/var/lib/freeswitch/db/sofia_reg_internal.db
/opt/freeswitch/var/lib/freeswitch/db/sofia_reg_internal.db
/opt/freeswitch/var/log/freeswitch/freeswitch.log
```

Alle von BBB-Prozessen auf dem Host genutzten regulären Dateien können auf diese Weise ermittelt werden:

26 man lsof: FD: r for read access; u for read and write access; w for write access

```
# for u in $(ps -axwwo user:16,pid|awk '{print $1}'|sort -n|uniq|egrep -v
  "$SUDO_USER|root|USER|post|daemon|message|system"|grep -E [a-zA-Z])
  do
  lsof -u $u|grep REG|grep -E '[0-9](u|w)'|awk -F/ -vOFS=# '{$1="";print $0}'|sed
  's/#/\//g'
  done
```

Anzumerken ist hier, dass die Log-Dateien des haproxy-Elternprozesses von „syslog” geschrieben werden und dieser Nutzer bei einer Selektion nicht ausgeschlossen werden kann.

Die so erzeugte Log-Liste enthielt keine Informationen darüber, wie die Dateien überwacht oder eingesehen werden können, deshalb wurden mit einer Erweiterung der obigen For-Schleife die Dateiendungen ermittelt.

```
# for u in ... done|awk -F. '{print $NF}'|sort -n|uniq
db
lock
log
pid
/tmp/2BW9L4 (deleted)
wt
0000000001
1 (deleted)
```

Dateien mit der Endung „lock” und „pid” wurden nach kurzer Prüfung nicht weiter beachtet, weil sie nur PID-relevante Informationen enthielten. Für Dateien mit der Endung .db, .log, .wt und .0000000001 musste herausgefunden werden, wie sie zur Laufzeit oder anderweitig ausgewertet werden können.

4.4.1 Log-Artefakte

Um den Dateityp der Log-Dateien zu ermitteln, wurde die obige For-Schleife mit ‚\.log‘ eingeschränkt und das Ergebnis in eine Datei namens „logs” umgeleitet. Aus dieser Log-Liste wurde in einer weiteren Schleife mit dem Befehl ‚file‘ der jeweilige Dateityp ermittelt. Das Ergebnis lieferte „ASCII text”, „HTML document” und „JSON data” zurück. Das sind i. d. R. Klartextdaten, die zur Laufzeit mit dem Befehl ‚tail‘ beobachtet oder mit einem Texteditor eingesehen werden können. Für eine erste Inhaltsübersicht wurde ‚tail‘ während des Starts eines Meetings verwendet. Auf diese Weise wird zwar zur Laufzeit der Inhalt der geschriebenen Daten angezeigt, allerdings sorgt diese Übersicht allenfalls für einen ersten Eindruck. Für eine tiefergehende Analyse ist sie nicht geeignet, da sie zu viele fortlaufende Informationen enthält.

```
# cat logs|tr '\n' ' '|xargs tail -f
```

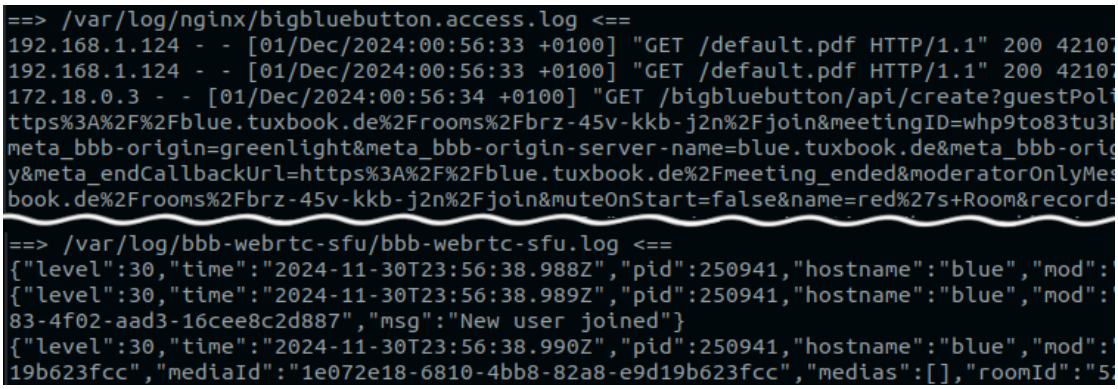

Bild 12 : Screenshot Ausgabe der Log-Dateien mit ‚tail' (gekürzt)

Im oberen Teil der Ausgabe in Bild 12 ist das Zugriffslog von nginx zu sehen, worin die IP-Adresse des BBB-Servers und die aufgerufene URL (default.pdf) zu erkennen ist. Ebenfalls werden Meeting-ID, Nutzername, Zeitpunkt der Teilnahme und Raumeigenschaften in der Abfrage für Greenlight angezeigt. Im unteren Teil ist die Log-Datei des WebRTC-SFU-Servers mit den Informationen zu Video und Audio des Nutzers im jeweiligen Raum zu sehen.

4.4.2 Datenbank-Artefakte im Dateisystem

Die Dateien mit Endung .wt wurden von ‚file' als Dateityp „dBase III DBT" deklariert.

```
# file /mnt/mongo-ramdisk/index/321--7524296707035955308.wt
  /mnt/mongo-ramdisk/index/321--7524296707035955308.wt: dBase III DBT, version
  number 0, next free block index 120897
```

Diese Dateien sind laut Dokumentation von WiredTiger (WT), der Speicher-Engine (storage engine) von MongoDB [42], erstellte und genutzte Daten. Sie enthalten die Kollektionen, Indizes, Journale und Transaktionen von MongoDB und können mithilfe des WT-eigenen Befehls ‚wt' ausgelesen werden. [43,44] Die Quellen der Software wurden bei github[27] heruntergeladen und kompiliert. Eine weitere Möglichkeit, auf die Inhalte der Dateien zuzugreifen, ist die grafische Benutzeroberfläche „Compass" von MongoDB[28]. Diese wurde zwar installiert, aber nicht genutzt, da die Ausgabe von ‚wt' ausreichend war.

27 https://github.com/wiredtiger/wiredtiger

28 https://www.mongodb.com/de-de/products/tools/compass

Beispiel zum Auslesen der WT-IDs und zugehöriger Kollektionen:

```
# wt dump -x table:_mdb_catalog|tail -n +7|awk 'NR%2==0 {print}'|xxd -r -p|
  bsondump --quiet|jq -r 'select(.|has("md"))|[.ident,.ns]|@tsv'|sort -k2
```

```
collection/23--7524296707035955308      admin.system.keys
collection/13--7524296707035955308      admin.system.version
collection/19--7524296707035955308      config.image_collection
collection/21--7524296707035955308      config.system.indexBuilds
collection/478--7524296707035955308     config.system.sessions
collection/17--7524296707035955308      config.transactions
collection/10--7524296707035955308      local.oplog.rs
collection/6--7524296707035955308       local.replset.election
collection/15--7524296707035955308      local.replset.initialSyncId
collection/4--7524296707035955308       local.replset.minvalid
collection/2--7524296707035955308       local.replset.oplogTruncateAfterPoint
collection/0--7524296707035955308       local.startup_log
collection/11--7524296707035955308      local.system.replset
collection/8--7524296707035955308       local.system.rollback.id
collection/114--7524296707035955308     meteor.annotations
collection/210--7524296707035955308     meteor.audio-captions
collection/304--7524296707035955308     meteor.auth-token-validation
```

Bild 13 : Screenshot Auszug von ‚wt dump' erzeugter Liste der WT-IDs und Kollektionen

Beispiel zum Auslesen einer Kollektion, hier „local.startup_log":

```
# wt dump -j collection/0--7524296707035955308|jq -a
```


```
  "WiredTiger Dump Version": "1 (10.0.2)",
  "table:collection/0--7524296707035955308": [
      "data": [
        {
          "key0": 1,
          "value0": "\u00fd\t\u0000\u0000\u0002_id\u0000\u0013\u0000\u0000\u0000blue-1731450130838\
93\u0001\u0000\u0000\u0002startTimeLocal\u0000\u0018\u0000\u0000\u0000Tue Nov 12 23:22:10.838\u0000
/meteor/bundle/mongo-ramdisk.conf\u0000\u0003net\u0000%\u0000\u0000\u0000\u0002bindIp\u0000\n\u0000
\u0000'\u0000\u0000\u0000\u0010oplogSizeMB\u0000\b\u0000\u0000\u0000\u0002replSet\u0000\u0004\u0000
\u0000\t\u0000\u0000\u0000disabled\u0000\bjavascriptEnabled\u0000\u0000\u0000\u0003setParameter\u00
000false\u0000\u0000\u0003storage\u0000\u00fe\u0000\u0000\u0000\u0002dbPath\u0000\u0013\u0000\u0000
000\u0001\u0000\u0003wiredTiger\u0000\u00b6\u0000\u0000\u0000\u0003collectionConfig\u0000\u001f\u00
```


Bild 14 : Screenshot Auszug eines von ‚wt dump' erzeugten Kollektionsinhaltes

Da die Daten in MongoDB direkt ausgelesen werden können und das Augenmerk nicht auf einer forensischen Analyse von WT-Dateien lag, erfolgte keine weitere Prüfung dieser Dateien.

Die Datei mit Endung .0000000001 wurde von ‚file' als „lif file" erkannt. Sie wurde wie die WT-Dateien im journal-Verzeichnis des „dbPath" von MongoDB (Verzeichnis der Datenbankdateien) gelistet. MongoDB erklärt in der Dokumentation [45], dass WT-Journale im Format „WiredTigerLog.<sequence>" geschrieben werden, wobei die Sequenz eine mit Nullen aufgefüllte Nummer ist, die bei 0000000001 beginnt. Die Journale enthalten Checkpoints, die zur Wiederherstellung der Datenkonsistenz z. B. nach einem unerwarteten Absturz genutzt

werden. Da auch diese Daten keine anderen Inhalte haben als jene, die direkt in der Datenbank ausgelesen werden können, wurde die Existenz des Journals zur Kenntnis genommen.

Die Dateien mit Endung .db wurden von ‚file' als Dateityp „SQLite 3.x database" deklariert. Mithilfe des SQLite-Browsers konnten sie analysiert werden.

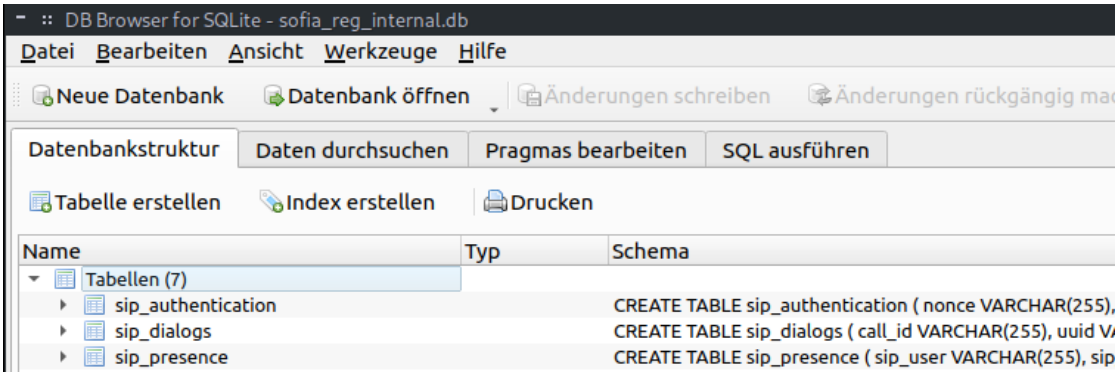

Bild 15 : Screenshot SQLite-Browser mit geöffneter .db-Datei (gekürzt)

4.4.3 Inhaltsartefakte

Auf die bisher beschriebene oder eine ähnliche Weise konnten auch andere, von den BBB-Komponenten erzeugte oder fortgeschriebene Artefakte ermittelt werden. Unter „ähnliche Weise" fällt der Einsatz des Befehls ‚find', mit dem gezielt nach Artefakten der BBB-Komponenten gesucht wurde. Das folgende Beispiel zeigt die Suche nach Verzeichnissen des Nutzers „bigbluebutton" als Grundlage für das Auffinden weiterer Artefakte unter Ausschluss der Verzeichnisse /proc und /run:

```
# find / -maxdepth 2 -type d \
  -not \( -path "/proc" -prune -o -path "/run" -prune \) \
  -user bigbluebutton
...
/var/bigbluebutton
/var/mediasoup

# find /var/bigbluebutton/* -maxdepth 0 -type d
...
/var/bigbluebutton/deleted
/var/bigbluebutton/events
/var/bigbluebutton/playback
/var/bigbluebutton/published
/var/bigbluebutton/recording
...
```

4.4.4 Nicht erfasste Artefakte

Beide Datenbank-Container (PostgreSQL, Redis) hatten ein Host-Verzeichnis gemountet und ihre Datenbank-Dateien darin gespeichert. Diese Artefakte hätten bei der Suche gefunden werden müssen. Die Gründe für das Nichtauffinden lagen zum Einen im Ausschluss des Nutzers mit der UID 70 (postgres), denn es wurde nur nach den dem Host bekannten Nutzern gesucht. Mit dem Wissen um den Postgres-Container wurde geschlussfolgert, dass im späteren Verlauf eine PostgreSQL-Datenbank untersucht werden wird und somit wurde dieser Nutzer bewusst vernachlässigt. Außerdem wurden bei der Suche die Verzeichnisse des Nutzers root ausgeschlossen. Das von UNIX übernommene Linux-Sicherheitsmodell DAC (Discretionary Access Control) stattet Applikationsprozesse i. d. R. mit den Rechten eines Applikationsnutzers (nonprivileged user) aus und versucht privilegierte Nutzer (privileged user, root) zu vermeiden. Der Grundsatz lautet, dass zur Minimierung von Angriffsvektoren [48] Prozesse und Dateien einer Applikation einem Applikationsnutzer gehören. [49,50] Darum stellt sich die Frage, warum nicht die Applikationsverzeichnisse des Hosts genutzt, sondern die Daten in das Home-Verzeichnis von root gespeichert werden.

4.5 Datenbank-Artefakte

Neben der in 4.4.2 erwähnten MongoDB sind zwei weitere Datenbanksysteme (DBMS) im Einsatz. Aus der Auflistung der Docker-Container in 4.2.1 geht hervor, dass ein Container namens „redis” und ein Container namens „postgres” läuft.

4.5.1 PostgreSQL

PostgreSQL ist eine objektrelationale SQL-Datenbank [47], die in der Image-Version 14.6-alpine3.17 vorlag. Der erste Schritt zur Erkundung von Daten im Postgres-Container war die Abfrage der Container-Informationen, um eventuelle im Container gemountete Verzeichnisse des Hosts zu finden.

```
# docker inspect postgres
...
"Mounts": [
```

```
  {
    "Type": "bind",
    "Source": "/root/greenlight-v3/data/postgres/14/database_data",
    "Destination": "/var/lib/postgresql/data",
    "Mode": "rw",
    "RW": true,
    "Propagation": "rprivate"
  }
], ...
```

Die von PostgreSQL genutzten Verzeichnisse befanden sich auf dem Host, wie die Ausgabe in Bild 16 bestätigt.

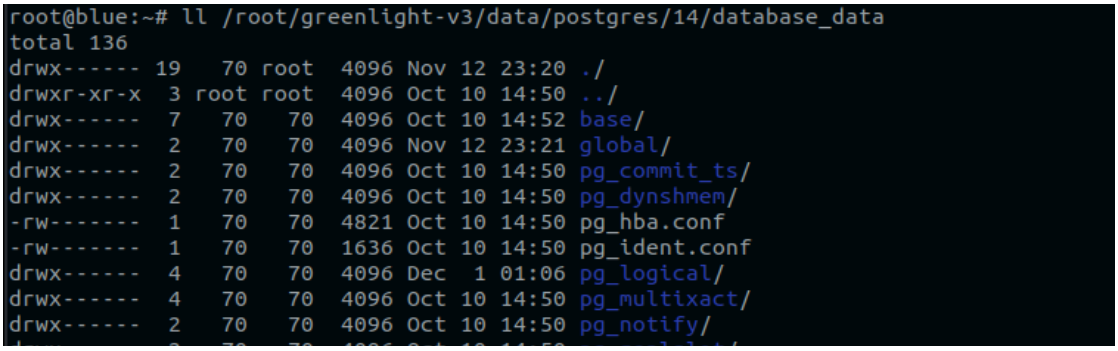

Bild 16 : Screenshot Inhalt des im Postgres-Container gemounteten Host-Verzeichnisses

Im Container selbst liefen die PostgreSQL-Prozesse unter dem Nutzer „postgres” mit der UID 70 (vgl. 4.2.2 Host-Prozesse).

```
Container:
b417bcf9adad:/# ps -ef
PID   USER     TIME  COMMAND
    1 postgres  0:28 postgres
   22 postgres  0:00 postgres: checkpointer
   23 postgres  0:08 postgres: background writer
...

b417bcf9adad:/# id postgres
uid=70(postgres) gid=70(postgres) groups=70(postgres),70(postgres)

b417bcf9adad:/# psql -U postgres
```

Es wurden die Datenbanken abgefragt, von denen eine DB namens „greenlight-v3-production” interessierte und deren Tabellen aufgelistet wurden (Bild 18).

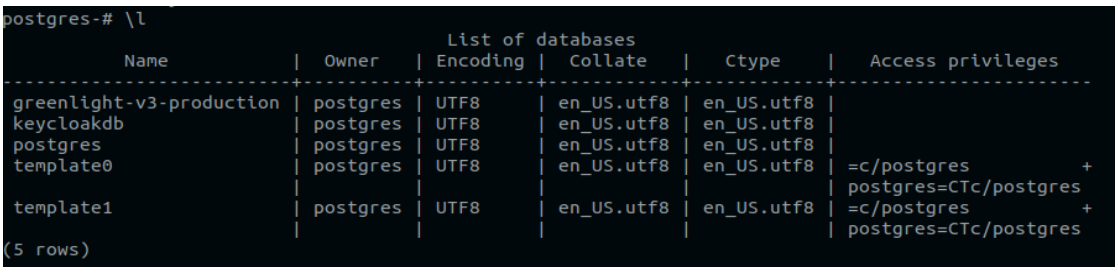

Bild 17 : Screenshot PostgreSQL-Datenbanken im Postgres-Container

```
postgres=# \c greenlight-v3-production
You are now connected to database "greenlight-v3-production" as user "postgres".
greenlight-v3-production=# \dt
                    List of relations
 Schema |              Name              | Type  |  Owner
--------+--------------------------------+-------+----------
 public | active_storage_attachments     | table | postgres
 public | active_storage_blobs           | table | postgres
 public | active_storage_variant_records | table | postgres
 public | ar_internal_metadata           | table | postgres
 public | data_migrations                | table | postgres
```

Bild 18 : Screenshot Auszug der Tabellen der DB „greenlight-v3-production” (gekürzt)

Um einen ersten Überblick zu erhalten, wurden die Daten aus der Tabelle „users” selektiert. Bild 19 zeigt die Grunddaten der Nutzer wie ID, Name, E-Mail-Adresse und ID-Provider (für eine bessere Übersicht mit DBeaver[29]). Mithilfe des Befehls ‚pg_dump‘ konnte ein SQL-Dump mit den Daten aller Tabellen erstellt, eingesehen und analysiert werden.

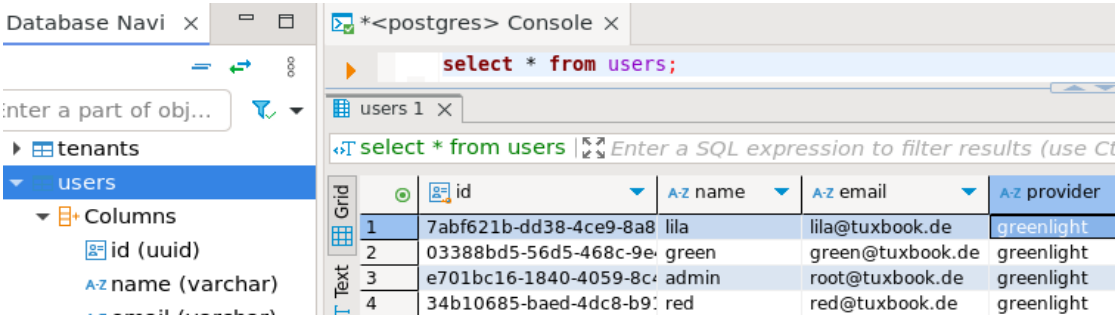

Bild 19 : Screenshot DBeaver: Auszug des Ergebnisses von SELECT * FROM users;

4.5.2 Redis

Redis zählt als Schlüssel-Wert-Datenbank (key-value store) zu den NoSQL-Datenbanksystemen. Sie verarbeitet die Daten im Hauptspeicher und wird deswegen zugleich als In-Memory-Datenbank bezeichnet. [46]. Zusätzlich nutzt Redis persistenten Festplattenspeicher für Backups (Dumps) und Logs in den Formaten RDB (Redis Database) und AOF (Append Only File). Obgleich ein redis-Container lief, wurde das auf dem Host laufende Datenbanksystem (DBMS) der Version 5:5.0.7-2ubuntu0.1 genutzt.

29 https://dbeaver.io/

Die Anzahl der Keys in der Redis-DB können mit dem Befehl ‚redis-cli --stat' angezeigt werden. Das war für die Ermittlung des genutzten DBMS (Container oder Host) sehr hilfreich.

```
# redis-cli --stat
------- data --------------- load -------------------- -
keys       mem      clients blocked requests            connections
702        1.84M    42      1       35417 (+0)          2681
709        1.94M    43      1       43758 (+4)          3330
717        1.99M    43      1       43829 (+71)         3330
...
```

Während des Startens eines Meetings wurden die Keys hochgezählt. Mit den Befehlen[30] der Redis CLI ließen sich Keys und Werte auslesen (hier verkürzt dargestellt):

```
# redis-cli KEYS '*'
  1) "recording:9541c1181c6d52c7880675764e400cb281a5..."
  2) "globalAuthor:a.xnw9E2DOHRWqxyDN"
  3) "author2sessions:a.ob4oKlUT2rXxwX4u"
  4) "session:s.85d394b3c8b066101ee1758e7f0eed4c"
...

# redis-cli type recording:9541c1181c6d52c7880675764e400cb281a5...
hash

# redis-cli hgetall recording:9541c1181c6d52c7880675764e400cb281a5...
 1) "chatEmphasizedText"
 2) "true"
 3) "meetingId"
 4) "9541c1181c6d52c7880675764e400cb281a53796-1733867416933"
...
 7) "senderId"
 8) "w_yrw8hw0wh4bs"
...
11) "message"
12) "Nix Hmm. Kadze!"
...
15) "module"
16) "CHAT"
...
```

In der obigen Beispielabfrage wurden zunächst der Datentyp[31] (hash) und anschließend die Werte der Keys abgefragt. Hervorgehoben ist die Meeting-ID („recording:..."), die senderId „w_yrw8hw0wh4bs" und die Chat-Antwort des Senders. Die Nutzer und ihre Eigenschaften werden im Key globalAuthor gespeichert und im Key mapper2author für Ereignisse abgebildet (mapped). Die Nutzerinformation zur Sender-ID konnte also mithilfe des „mapper-Keys" abgefragt werden.

30 https://redis.io/docs/latest/commands/

31 https://redis.io/docs/latest/develop/data-types/

```
# redis-cli get mapper2author:w_yrw8hw0wh4bs
"\"a.xnw9E2DOHRWqxyDN\""

# redis-cli get globalAuthor:a.xnw9E2DOHRWqxyDN
"{\n    \"colorId\": 57,\n    \"name\": \"black\",\n    \"timestamp\":
 1733867422614\n}"
```

Das Ergebnis der Abfrage zeigt demnach die Chat-Antwort des Nutzers „black". Bild 20 zeigt einen Screenshot der „Unterhaltung".

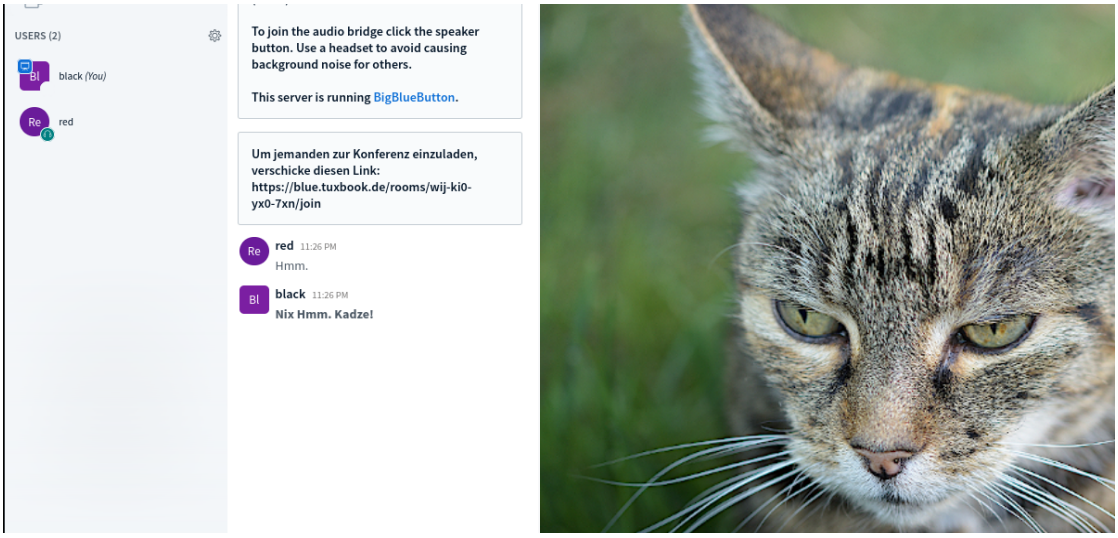

Bild 20 : Screenshot einer Unterhaltung im Chat

Mithilfe der Redis CLI konnten alle Dokumente aus Redis extrahiert und analysiert werden.

4.5.3 MongoDB

Wie in 4.4.2 erwähnt, war außerdem eine MongoDB im Einsatz. Hier lag sie in Version 4.4.29 vor. MongoDB ist ein NoSQL-Datenbanksystem, das Informationen in Dokumenten speichert, wobei Dokumente die Informationen als Feld-Wert-Daten enthalten und in Kollektionen („collections") gespeichert werden. [23] Dabei verwendbare Dokumentenformate sind JSON, BSON (binäres JSON) oder XML. NoSQL-DBMS sind nicht-relationale, Datenbankschema-freie Systeme mit flexiblen Datenstrukturen, die skalierbare Anwendungen in Echtzeitverarbeitung unterstützen. [24,25]

Die Daten konnten mithilfe der „MongoDB shell” abgefragt werden. Allerdings waren sie nur während eines Meetings verfügbar. Im Folgenden ist die Verbindung mit dem DBMS und die Abfrage der Datenbanken zu sehen.

```
# mongo mongodb://127.0.1.1:27017/
MongoDB shell version v4.4.29
...
rs0:PRIMARY> show dbs
admin   0.000GB
config  0.000GB
local   0.002GB
meteor  0.002GB
```

Da BBB, wie in 3.1 Frontend und Backend beschrieben, Meteor nutzt, wurde sich auf die Datenbank „meteor” konzentriert und die darin vorhandenen Tabellen abgefragt (im Folgenden verkürzt dargestellt):

```
rs0:PRIMARY> use meteor
switched to db meteor
rs0:PRIMARY> show tables
annotations
audio-captions
auth-token-validation
...
user-reaction
users
users-infos
users-persistent-data
...
```

Mit dem ‚find()‘ Befehl konnten z. B. die Nutzerdaten abgefragt werden:

```
rs0:PRIMARY> db.users.find().pretty()
{
  "_id" : "Zo44RS6SSKgzja2i6",
  "meetingId" : "9541c1181c6d52c7880675764e400cb281a5...",
  "userId" : "w_yrw8hw0wh4bs",
  "authToken" : "avlaamhl9iff",
...
  "intId" : "w_yrw8hw0wh4bs",
  "locked" : true,
  "loggedOut" : false,
  "mobile" : false,
  "name" : "black",
}
```

Mithilfe von ‚mongodump‘ konnten die Daten zur Laufzeit aus der Meteor-Datenbank extrahiert und anschließend mit ‚bsondump‘ in ein lesbares JSON-Format konvertiert und analysiert werden.

```
# mongodump --host="127.0.1.1:27017" --db=meteor --out /tmp/
# cd /tmp/meteor
# for i in $(ls *.bson|cut -d. -f1); do bsondump --pretty --outFile $i.json
  $i.bson; done
```

4.5.4 SQLite

In der BBB-Dokumentation findet sich kein Hinweis auf die Verwendung einer SQLite-Datenbank. Die in 4.4.2 erwähnte DB-Datei ist in einem FreeSWITCH-Verzeichnis gespeichert und ihre Tabellenstruktur lässt darauf schließen, dass darin die Telefonanrufe erfasst werden. Die Möglichkeit, einem Meeting telefonisch beizutreten, bestand in der Laborumgebung nicht, demzufolge befanden sich in der SQLite-Datenbank keine Daten.

4.6 Erfassung der Kommunikation mit dem Client

Mit den Nutzern wurden verschiedene Meetingaktivitäten durchgeführt und die Kommunikation mithilfe der Kali-VM aufgezeichnet. Bild 21 zeigt eine Client-Anfrage beim Login-Prozess und Bild 22 die Antwort des Servers. Auf diese Weise konnte die Kommunikation zwischen Client und Server analysiert werden.

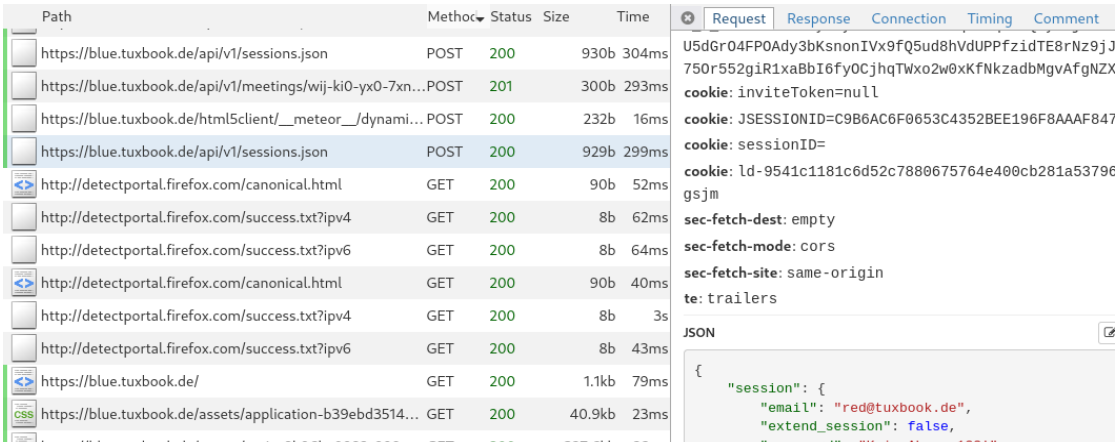

Bild 21 : Screenshot mitmweb: Mitschnitt des Logins; Client-Request (gekürzt)

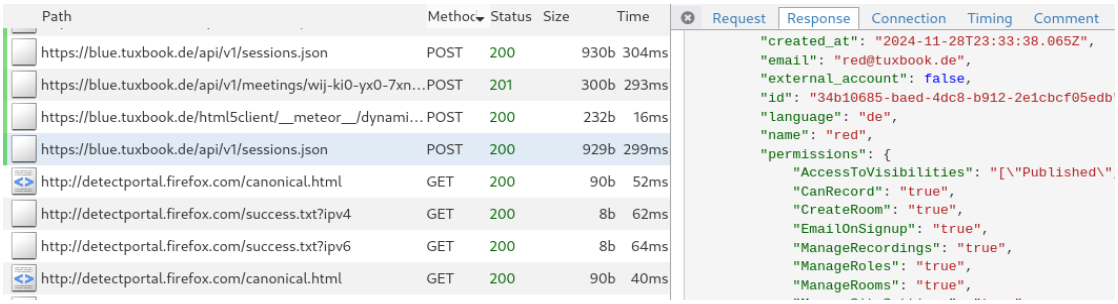

Bild 22 : Screenshot mitmweb: Mitschnitt des Logins; Server-Response (gekürzt)

5 Datenschutzrechtliche Einordnung der Verarbeitung

5.1 Datenschutzrechtliche Begriffe

5.1.1 Personenbezogene Daten

Personenbezogene Daten werden in Art. 4 Nr. 1 der Datenschutz-Grundverordnung (DSGVO) wie folgt definiert: [30]

1. „personenbezogene Daten“ alle Informationen, die sich auf eine identifizierte oder identifizierbare natürliche Person (im Folgenden „betroffene Person“) beziehen; als identifizierbar wird eine natürliche Person angesehen, die direkt oder indirekt, insbesondere mittels Zuordnung zu einer Kennung wie einem Namen, zu einer Kennnummer, zu Standortdaten, zu einer Online-Kennung oder zu einem oder mehreren besonderen Merkmalen, die Ausdruck der physischen, physiologischen, genetischen, psychischen, wirtschaftlichen, kulturellen oder sozialen Identität dieser natürlichen Person sind, identifiziert werden kann;

Die bei einer Videokonferenz verarbeiteten personenbezogenen Daten beschränken sich nicht auf Namen, E-Mail-Adresse, IP-Adresse oder Telefonnummer der Teilnehmer. Die Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder (DSK) ordnet in der „Orientierungshilfe Videokonferenzsysteme” auch folgende Informationen der Teilnehmer den personenbezogenen Daten zu: [32]

- Inhaltsdaten wie inhaltliche Äußerungen, Anzeige von Bildschirminhalten, Ton und Bild der Teilnehmer und ggf. ihres Umfeldes (Arbeits- oder Wohnraum),
- Metadaten wie Daten zur Kommunikationsdurchführung, Kontakte, Arbeitszeiten oder -leistung,
- sonstige Daten mit Personenbezug in Textbeiträgen oder Dokumenten,
- Daten von Personen aus dem Umfeld während einer Konferenz

Mit dem relativen Grundverständnis des Personenbezugs kann auch eine alphanumerische Meeting-ID zu einem personenbezogenem Datum werden, nämlich „für denjenigen, der bei vernünftiger Betrachtung über Mittel verfügt, die es ermöglichen, sie einer bestimmten Person zuzuordnen". [31] Das trifft auf den Betreiber von BBB zu, dessen Mittel in der Verknüpfung gespeicherter Daten besteht, was Komponenten-übergreifend genutzt wird.

5.1.2 Besondere Kategorien personenbezogener Daten

Bei einer Videokonferenz können mittels Webcam und Mikrofon sowohl das Gesichtsbild als auch die Stimme der Teilnehmer übertragen werden und so stellt sich zwangsläufig die Frage nach der Verarbeitung biometrischer Daten, die in Art. 9 Abs. 1 DSGVO den besonderen Kategorien personenbezogener Daten zugeordnet werden. Biometrische Daten werden in Art. 4 Nr. 14 DSGVO wie folgt definiert: [30]

14. „biometrische Daten" mit speziellen technischen Verfahren gewonnene personenbezogene Daten zu den physischen, physiologischen oder verhaltenstypischen Merkmalen einer natürlichen Person, die die eindeutige Identifizierung dieser natürlichen Person ermöglichen oder bestätigen, wie Gesichtsbilder oder daktyloskopische Daten;

Physiologische Merkmale sind neben dem Gesicht z. B. der Fingerabdruck oder das Irismuster. Zu verhaltensbedingten Merkmalen zählen das Schreibverhalten, die Lippenbewegung oder die Stimme. [34] Folgt man der Einschätzung der DSK im „Positionspapier zur biometrischen Analyse", handelt es sich bei Gesichtsbildern von Teilnehmern (und Personen im Umfeld) um biometrische Daten, wenn biometrische Eigenschaften des Gesichts für die Erstellung biometrischer Templates oder strukturierter Sammlungen verarbeitet werden können. [33] Im Rahmen eines automatisierten Verfahrens ließe sich das Gesichtsbild eindeutig einer Person zuordnen und ermöglichte dadurch die eindeutige Identifizierung einer natürlichen Person. Dies korrespondiert mit Erwägungsgrund 51 Satz 3 DSGVO:

Die Verarbeitung von Lichtbildern sollte nicht grundsätzlich als Verarbeitung besonderer Kategorien von personenbezogenen Daten angesehen werden, da Lichtbilder nur dann von der Definition des Begriffs „biometrische Daten“ erfasst werden, wenn sie mit speziellen technischen Mitteln verarbeitet werden, die die eindeutige Identifizierung oder Authentifizierung einer natürlichen Person ermöglichen.

Die Einbeziehung dieser Definition ist deshalb relevant, weil die Verarbeitung besonderer Kategorien personenbezogener Daten eine Datenschutz-Folgeeinschätzung und andere Schutzmaßnahmen bedarf (vgl. Art. 9 DSGVO).

5.1.3 Verarbeitung

Eine Verarbeitung personenbezogener Daten normiert die DSGVO in Art.l 4 Nr. 2 wie folgt:

2. „Verarbeitung“ jeden mit oder ohne Hilfe automatisierter Verfahren ausgeführten Vorgang oder jede solche Vorgangsreihe im Zusammenhang mit personenbezogenen Daten wie das Erheben, das Erfassen, die Organisation, das Ordnen, die Speicherung, die Anpassung oder Veränderung, das Auslesen, das Abfragen, die Verwendung, die Offenlegung durch Übermittlung, Verbreitung oder eine andere Form der Bereitstellung, den Abgleich oder die Verknüpfung, die Einschränkung, das Löschen oder die Vernichtung;

Näher erläutert wird dies in 5.3 Einordnung der Verarbeitung.

5.2 Zusammenfassung ermittelter Daten

Die Tabelle 1 fasst die ermittelten personenbezogenen Daten als Datenkategorien zusammen und führt zugehörige Daten auf. Die Darstellung orientiert sich an den in Art. 30 und Art. 33 DSGVO (Verzeichnisse zu Verarbeitungstätigkeiten, Meldungen zu Datenschutzverletzungen) geforderten Angaben von Datenkategorien (Kategorien von personenbezogenen Daten). Zum Teil überschneiden sich die Daten der einzelnen Kategorien, was der redundanten Speicherung und vielfachen Verknüpfung der Daten geschuldet ist. Der Datenumfang

umfasst mit Ausnahme der telefonischen Teilnahme die Nutzung sämtlicher Meetingfunktionen. Die Bezeichnung Meeting ist mit Konferenz gleichzusetzen.

Datenkategorie	**Daten**	**Artefakt**
Meetingdaten	• Meeting-ID (Raum-ID), Meeting-Name, hochgeladene Dateien, Benachrichtigungen, Speicherungen, Transkriptionen • Notiz-ID, Chat-ID • Eigenschaften Aufzeichnungen • Moderator-ID, genutzte Module, User-Rollen, Nutzer-ID[I] • Moderator- und Beitrittspasswort • Session- und Auth-Token • Verzeichnis- und Dateinamen • URL, Links zu Präsentationen • Whiteboardnutzung • Zeitstempel: erstellt, aktualisiert, beendet, Gültigkeitsstatus	MongoDB[I] Redis PostgreSQL Log-Dateien Inhaltsdateien[II] int. Net-Komm.
Ereignisse (Events)	• Bildschirmfreigabe, Beitritt und Verlassen der Teilnehmer, Chat- und Notizennutzung, Hochladen von Dateien, Nutzung von Emojis • Zeitstempel: Start, Aktualisierung	Redis int. Net-Komm.
Notizen	• Notiz-ID, Meeting-ID, Nutzer-ID, Session-ID • Text	MongoDB Redis Inhaltsdateien int. Net-Komm.
Chat	• Chat-ID, Meeting-ID, Nutzer-ID (Verfasser), Nutzer-Name, Nutzer-Rolle • Nachricht • Zeitstempel	MongoDB Redis Log-Dateien Inhaltsdateien int. Net-Komm.
Umfragen	• Meeting-ID, Nutzer-ID (Ersteller, Teilnehmer) • Umfrageinhalt, Abstimmungsdaten • Zeitstempel	MongoDB Inhaltsdateien int. Net-Komm.
Präsentationen	• Meeting-ID, Nutzer-ID, Auth-Token • Name, Dateiname, Inhalte (Slides) • Zeitstempel	MongoDB Log-Dateien Inhaltsdateien int. Net-Komm.

Datenkategorie	**Daten**	**Artefakt**
Nutzerdaten	• Nutzer-ID, Nutzername, Passwort, PIN, Nutzerrolle, Sprache, E-Mail-Adresse, ID-Provider, IP-Adresse, Avatar • Meeting-ID, Moderatorstatus, Validierungsstatus, Teilnahmestatus (pausierend, beendet), Verifizierungsstatus, Präsentattionsstatus, Gaststatus, Account-Aktivierung • mobile Nutzung, Webcam-Hintergrund • Audioeinstellungen • Zeitstempel (letzter) Login, Logout, Erstellung Account, letzte Session, aktualisiert, Teilnahme, Notizen, Chat, Ereignisse, Präsentation	MongoDB Redis PostgreSQL Log-Dateien Inhaltsdateien int. Net-Komm.
Inhaltsdaten	Meeting-ID, Nutzer-ID, Zeitstempel für sämtliche Inhalte der • Bildschirmfreigabe • Webcams der Teilnehmer (WEBM) • Aufzeichnungen (WEBM) • Präsentationen als Text, SVG und im ursprünglichen Format • Chats • Notizen • Umfragen • Audiodaten (OPUS)	/var/bigbluebutton/ int. Net-Komm.
Aufzeichnungen	Meeting-ID, Aufzeichnungs-ID, Dateiname, URL	PostgreSQL Log-Dateien Inhaltsdateien int. Net-Komm.
Videodaten (Webcam)	Meeting-ID, Nutzer-ID, Nutzername, PIN, Stream-ID, Streamname, Device- und Kamera-ID, Media-ID, Media-Session-ID	MongoDB Log-Dateien Inhaltsdateien (bei Aufzeichnung) int. Net-Komm.
Audiodaten	• Meeting-ID, Nutzer-ID, Nutzername, Telefonnummer • Dateiname • Teilnahmestatus (in Konferenz, Anruf gestartet), • Audioeigenschaften (stumm, sprechend) • Media-ID, Media-Session-ID • Zeitstempel Beginn und Ende	MongoDB Inhaltsdateien (bei Aufzeichnung) int. Net-Komm.

Datenkategorie	Daten	Artefakt
Streamingdaten	analog Video- und Audiodaten ohne Aufzeichnung	
Bildschirminhalte	Meeting-ID, Freigabe-ID, Stream-ID Dateiname, Zeitstempel	MongoDB Log-Dateien Inhaltsdateien int. Net-Komm.
Browserdaten	User-Agent inkl. Eigenschaften, Betriebssystem, Sprache, Referrer	Log-Dateien int. Net-Komm.
Sessiondaten	Session-ID, Cookies, Auth und Session-Token	Log-Dateien int. Net-Komm.
Kommunikationsdaten	IP-Adresse, URLs, Meetinglinks, E-Mail, Zeitstempel	Log-Dateien int. Net-Komm.
Authentifizierung	• E-Mail-Adresse, Nutzername, Passwort • E-Mail: URL für Verifizierung und Passwortreset	PostgreSQL E-Mail-Postfach int. Net-Komm.

Tabelle 1 : Übersicht der ermittelten personenbezogene Daten

[I] Nutzer-ID beinhaltet „mapped" und „unmapped" User-IDs in DBs und Dateien

[II] Dateien im Dateisystem, die nutzerbezogene Daten wie Audio, Video, Text enthalten

[III] interne TCP-, UDP- und Websocket-Kommunikation

5.3 Einordnung der Verarbeitung

5.3.1 Allgemeines

Die Zusammenfassung in Tabelle 1 detailliert die Auflistung der DSK in 5.1.1 und spiegelt den Umfang der personenbezogenen Daten wider. Auf dem BBB-Server findet ein Großteil der Verarbeitungsformen statt. Bei der Anmeldung wird z. B. die IP-Adresse erhoben und die E-Mail-Adresse erfasst und gespeichert. Eine Organisation (Strukturierung) erfahren die Daten z. B. durch die Ordnung in den Datenbanken oder die (inhaltliche) Formatierung von Daten (z. B. in Logs). Die Daten werden den Teilnehmern durch Übermittlung offengelegt (Daten von Teilnehmern werden an Teilnehmer übertragen). Sollte keine

Einschränkung für die Meetings existieren, kann es zu einer Verbreitung oder auch der Bereitstellung von Aufzeichnungen kommen. [vgl. 52] Die Daten werden im Rahmen der Meetingteilnahme oder -aktivitäten verwendet, gespeichert und nach Beendigung gelöscht (z. B. Streamingdaten).

5.3.2 Art der Speicherung - Artefakteigenschaften

Aufgrund der Nutzung verschiedenster, in 3.1 beschriebener Komponenten, erfolgt über zahlreiche Protokolle und Ports ein Austausch aller erfassten Daten. Diese Daten der internen TCP-, UDP- und Websocket-Kommunikation sind von umfangreicher, jedoch flüchtiger Natur.

Durch die Synchronisation der Inhalte des Arbeitsspeichers mit den Inhalten auf der Festplatte sind Daten der In-Memory-DB von Redis nicht flüchtig. Ebenso wenig flüchtig sind die Daten in der MongoDB. Zwar können sie nur zur Laufzeit eines Meetings direkt aus der DB ausgelesen werden, allerdings werden die Kollektionen in den WiredTiger-Dateien gespeichert (siehe 4.4.2 Datenbank-Artefakte im Dateisystem) und können für eine Datenwiederbeschaffung herangezogen werden. Die PostgreSQL-DB hält die Grunddaten, wie Name, Passwort, E-Mail und Konfigurationen von Räumen resistent vor.

Die Daten der Log-Dateien sind resistent, sowohl durch den Einsatz archivierender Logrotate-Routinen (z. B. beim Webserver), als auch durch die Tatsache, dass einige Log-Dateien erst nach einem bestimmten Zeitraum oder beim Neustart von BBB gelöscht werden.

Die Inhaltsdateien verbleiben auf der Festplatte. Wird das Meeting nicht aufgezeichnet, sind jedoch die Audio- und Videodaten von Mikrofon und Webcam flüchtig, d. .h. sie werden nur während des Meetings als Streamingdaten verarbeitet (verwendet, übermittelt).

Es wurde festgestellt, dass die gespeicherten Daten für eine Aktivitätsrekonstruktion herangezogen werden können.

5.3.3 Verarbeitung biometrischer Daten

Aus Gesichtsbildern von Teilnehmern ist mithilfe von Gesichtserkennungssoftware eine eindeutige Identifizierung möglich. Das zeigt eine Studie, in der mit Gesichtserkennungsalgorithmen aus Bildcollagen von Meetingteilnehmern und dem Abgleich mit Bildern aus den Sozialen Medien eine Wiedererkennungsrate von 80% erreicht wurde. Zudem konnten die Forscher die extrahierten Einzelbilder in Vektoren darstellen, das Geschlecht, die Namen und auch das Alter der Teilnehmer herausfinden. [35] Einerseits wird Software entwickelt, um Gesichtserkennung in webbasierten Videokonferenzsystemen per Javascript einzubinden. [37] Andererseits wird Software entwickelt, um Gesichts- und Stimmenerkennung zu erschweren oder zu verhindern [36]. Dies lässt wiederum den Schluss zu, dass bei Videokonferenzen biometrische Daten verarbeitet werden können, da die Voraussetzungen dafür grundsätzlich erfüllt wären. Nichtsdestotrotz muss für eine Zuordnung zu den biometrischen Daten eine automatisierte Verarbeitung auf dem BBB-System möglich sein. Eine entsprechende Software war nicht vorhanden. Insofern kann eine Verarbeitung biometrischer Daten in diesem Fall verneint werden.

5.3.4 Auffälligkeiten

Während der Analyse fielen bereits folgende Punkte auf:

- Die Verzeichnisse der und die Inhaltsdateien selbst gehörten zwar dem Nutzer „bigbluebutton”, sind aber für Dritte (other) lesbar.
- Auf der Webseite gelöschte Aufzeichnungen wurden vom Verzeichnis „published” ins Verzeichnis „deleted” verschoben und nicht gelöscht; zudem verbleiben die Daten im Verzeichnis „raw”.
- Die DB-Dateien der Container wurden zwar als DB-Nutzer jedoch im Home-Verzeichnis von root gespeichert. Das erschwerte das Auffinden einiger Artefakte. (vgl. 4.4.4 Nicht erfasste Artefakte).

- Die Container wurden nicht für den Rootless-Modus[32] konfiguriert.
- Die interne Kommunikation findet in der Standardkonfiguration unverschlüsselt statt.
- Die IP-Adresse der Clients wurde nicht als Hash gespeichert.
- Die redundante Speicherung der Informationen in den Komponenten erschwert das Löschen und die Änderung einzelner Datensätze.
- Die URLs von Präsentationen und Aufzeichnungen konnten ohne Authentifizierung aufgerufen werden.

32 https://docs.docker.com/engine/security/rootless/

6 Einordnung in das Standard-Datenschutzmodell

Das Standard-Datenschutzmodell (SDM) wird vom AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder herausgegeben und von den Aufsichtsbehörden gemeinsam beschlossen. [55] Als „Methode zur Datenschutzberatung und -prüfung" böte es geeignete Maßnahmen, um die rechtlichen Anforderungen der DSGVO in technische und organisatorische Maßnahmen zu überführen, erläutern die Verfasser. Durch einen Referenzmaßnahmen-Katalog unterstützt das SDM „die Transformation abstrakter rechtlicher Anforderungen in konkrete technische und organisatorische Maßnahmen", trägt mit Standardisierungen zu einem transparenten und nachvollziehbaren System bei und stellt eine Methode zur Risikominimierung der Rechte und Freiheiten natürlicher Personen bei der Verarbeitung personenbezogener Daten bereit. [56]

Für eine Einordnung oder Gegenüberstellung der Verarbeitungen und Maßnahmen in Bezug auf das SDM, muss zuerst geklärt werden, welchen Zweck die Maßnahmen erfüllen sollen. Dies wird durch die Gewährleistungsziele des SDM erläutert. Die Gewährleistungsziele basieren auf den Grundsätzen für die Verarbeitung personenbezogener Daten des Art. 5 DSGVO. Das SDM dient somit dem Abgleich der datenschutzrechtlichen Anforderungen wie z. B. der Rechtmäßigkeit der Verarbeitung, des Zwecks oder der Ist- und Soll-Zustände unter dem Aspekt der in Art. 32 DSGVO geforderten Sicherheit der Verarbeitung.

Das SDM führt zwar an den Gewährleistungszielen ausgerichtete Maßnahmen auf, allerdings ist der „Anhang Referenzmaßnahmen-Katalog" nicht im SDM enthalten. Erst eine ältere Version offenbart, dass auf der Webseite des Landesbeauftragten für Datenschutz und Informationsfreiheit Mecklenburg-Vorpommern ein Repository mit Bausteinen des Maßnahmekatalogs[33] gepflegt werden soll. Diese Bausteine differieren sowohl in der Versionierung (SDM Version 3.1 vs. Bausteine für SDM Version 2.0) als auch bei den Formulierungen. Während das SDM durchgängig die Begriffe der Gewährleistungsziele verwendet, orientieren sich die Bausteine teilweise an den Begriffen des Bundesdatenschutzge-

33 https://www.datenschutz-mv.de/datenschutz/datenschutzmodell/ → Maßnahmekatalog

setzes (BDSG) vor Inkrafttreten der DSGVO, listen die Gewährleistungsziele lediglich zu Beginn auf und lassen eine konkrete Überführung der Gewährleistungsziele in technische Maßnahmen missen. Die Verweise zum BSI laufen ins Leere, da das BSI für den technischen Datenschutz auf das SDM verweist. Diese Vorgehensweise erscheint inkonsistent und wenig hilfreich für eine Überführung der Anforderungen auf konkrete Umsetzungen, mit denen sich Administratoren und Softwareentwickler erfahrungsgemäß schwer tun.

Im Folgenden werden als Sollzustand die im SDM aufgeführten generischen oder typischen technischen Maßnahmen zur Erreichung der Gewährleistungsziele dem ermittelten Istzustand der BBB-Installation gegenübergestellt. Da organisatorische Maßnahmen größtenteils dem Verantwortlichen und nicht dem Hersteller zuzuordnen sind, werden sie hier vernachlässigt. Einige Maßnahmen und Zustände wiederholen sich, wurden aber der Vollständigkeit halber aufgeführt. Der Gegenüberstellung anschließend werden konkrete Maßnahmen aufgeführt (auf eine Wiederholung einzelner Maßnahmen wurde hier verzichtet).

6.1 Gewährleistungsziel Verfügbarkeit

Verfügbarkeit fordert den für eine Verarbeitung notwendigen Zugriff auf die Daten. Sie müssen auffindbar, darstellbar und wiederherstellbar sein. Hierunter fallen auch Lastgrenzen - das System muss für die Belastung, die durch Verarbeitungen entstehen, geeignet sein. [58] Neben der Wiederherstellung von Daten aus DB-Dumps oder Backups (unbeabsichtigten Verlust) zählt hierzu auch die Vermeidung korrupter Daten (unbeabsichtigte Schädigung), der erforderliche Zugriff auf das Dateisystem oder die Einbindung von Monitoringsystemen[34].

B bezeichnet das Backend, F ist die Abkürzung für Frontend.

Sollzustand Maßnahmen SDM D1.1	**Istzustand**
Anfertigung von Sicherheitskopien von Daten, Prozesszuständen, Konfigurationen, Datenstrukturen, Transaktionshistorien u. ä.	• B: Synchronisierung der In-Memory-DB Redis auf Festplatte • B: MongoDB Wiederherstellung via Journal • B: angepasste Konfigurationen können

34 https://bigbluebutton-exporter.greenstatic.dev/

Sollzustand Maßnahmen SDM D1.1	**Istzustand**
	z.T. erhalten werden, Sicherung manuell
Schutz vor äußeren Einflüssen (Schadsoftware, Sabotage, höhere Gewalt)	B: Firewall aktiv, betreiberabhängig
Dokumentation der Syntax der Daten	• nicht vorhanden • Developer-Dokumentation für DBs, Meteor, React, Etherpad, Kurento über Dokumentation bei Hersteller
Redundanz von Hard- und Software sowie Infrastruktur	nicht zutreffend (betreiberabhängig)
Umsetzung von Reparaturstrategien und Ausweichprozessen	B: HTML5-Server läuft mit mehreren Instanzen
Erstellung eines Notfallkonzepts zur Wiederherstellung einer Verarbeitungstätigkeit	nicht zutreffend (betreiberabhängig)

Tabelle 2 : Soll- und Istzustand für das Gewährleistungsziel Verfügbarkeit

Konkrete Maßnahmen wären:

- Implementierung oder Anpassung von SELinux[35]- oder AppArmor[36]-Policies
- Bereitstellung einer Komponenten-übergreifenden Dokumentation der Daten-Syntax
- Bereitstellung eines Debug-Modus ohne manuelle Eingriffe in die Konfiguration, z. B. durch einen Startparameter

6.2 Gewährleistungsziel Integrität

Die Integrität fordert die Einhaltung der Spezifikationen von Prozessen und Systemen sowie die Vollständigkeit, Richtigkeit und Unversehrtheit der Daten. Eine Herausforderung ist hierbei die Fehler- und Diskriminierungsfreiheit vor allem bei der Anwendung von KI-Verfahren. [59] Ein Beispiel für unrichtige Daten wäre die Zuordnung einer Chatnachricht zu einem Teilnehmer, obwohl dieser gar nichts geschrieben hatte.

35 https://www.redhat.com/en/topics/linux/what-is-selinux

36 https://apparmor.net/

Sollzustand Maßnahmen SDM D1.2	**Istzustand**
Einschränkung von Schreib- und Änderungsrechten	• B: Verzeichnisse / Dateien nur durch Besitzer änderbar • B: Daten in Redis und MongoDB durch nonpriviledged Nutzer editierbar • B: Konfigurationsdateien sind ausführbar
Einsatz von Prüfsummen, elektronischen Siegeln und Signaturen in Datenverarbeitungsprozessen gemäß eines Kryptokonzepts	nicht erkennbar
dokumentierte Zuweisung von Berechtigungen und Rollen	• F: Rechte und Rollen für Teilnehmer konfigurierbar • B: Rechte und Rollen für DBs und Dateisystem teilweise umgesetzt
Löschen oder Berichtigen falscher Daten	• B: in DBs und Inhaltsdateien manuell oder automatisiert und eingeschränkt möglich, keine erkennbare Routine • F: nicht vorhanden (z. B. für den Moderator, Admin)
Härten von IT-Systemen, so dass diese keine oder möglichst wenige Nebenfunktionalitäten aufweisen	• OS betreiberabhängig • B: nur für den BBB-Betrieb notwendige Software installiert • B: Update-Routine für einzelne Komponenten unbekannt, z. T. veraltete Versionen im Einsatz • B: Docker-Container nicht für Rootless Modus konfiguriert, auf Host-Ebene nur für root berechtigt (keine Nutzung durch nonpriviledged Nutzer möglich)
Prozesse zur Aufrechterhaltung der Aktualität von Daten	nicht vorhanden, Notwendigkeit unbekannt
Prozesse zur Identifizierung und Authentifizierung von Personen und Gerätschaften	• B: Authentifizierung über Linux-Sicherheitsmodell • B: für NoSQL-DBs in der Standardkonfiguration nicht vorhanden • F: Authentifizierung mit E-Mail-Adresse und Passwort
Festlegung des Sollverhaltens von Prozessen und regelmäßiger Durchführung von Tests zur Feststellung und Dokumentation der Funktionalität, von Risiken sowie Sicherheitslücken und Nebenwirkungen von Prozessen	teilweise betreiberabhängig teilweise auf der BBB-Webseite dokumentiert
Festlegung des Sollverhaltens von Abläufen bzw. Prozessen und regelmäßiger Durchführung von Tests zur Feststellbarkeit bzw. Feststellung der	teilweise betreiberabhängig B: Monitoring kann implementiert werden

Sollzustand Maßnahmen SDM D1.2	**Istzustand**
Ist-Zustände von Prozessen	
Schutz vor äußeren Einflüssen (Spionage, Hacking)	• B: Firewall aktiv • Spionage- / Hackangriff nicht ausgeführt

Tabelle 3 : Soll- und Istzustand für das Gewährleistungsziel Integrität

Bild 23 zeigt die in Zeile 1 erwähnte Manipulation von Nutzerdaten in MongoDB durch einen nicht privilegierten Nutzer. Hier wurde der Wert des „authToken" von „pigbp672jyqo" auf „4711" geändert. Eine Änderung des Wertes für „raiseHand" von false auf true hatte den Effekt, dass der Nutzer „red" im Meeting die Hand hob. Eventuelle NoSQL-Injections wurden nicht getestet.

Bild 23 : Screenshot Manipulation des „authToken" Felds im Dokument des Nutzers „red"

Konkrete Maßnahmen wären:

- Änderung der Besitzer von Verzeichnissen und Dateien auf den jeweiligen Applikationsnutzer, nötigenfalls Erstellung neuer Nutzer
- Steuerung von notwendigen komponentenübergreifenden Zugriffsrechten auf Verzeichnisse und Dateien über Gruppenzugehörigkeiten
- Einschränkung der Lese- und Schreibrechte von Verzeichnissen und Dateien (Inhaltsdateien, Konfigurationen, ausführbare Dateien) auf die jeweiligen Applikationsnutzer und -gruppen
- Dokumentation der zugewiesenen Berechtigungen

- Bereitstellung einer Routine zum Löschen einzelner Datensätze
- Bereitstellung der Möglichkeit, Sicherheitsupdates für die Komponenten einzuspielen
- standardmäßige Verwendung von Passwörtern für Datenbanken
- siehe Integrität

6.3 Gewährleistungsziel Vertraulichkeit

Die Vertraulichkeit stellt die Anforderung des Ausschlusses der Kenntnisnahme und Nutzung personenbezogener Daten durch eine unbefugte Person. Unbefugt ist jeder, der „keinerlei inhaltlichen Bezug zu einer Verarbeitungstätigkeit oder zu der jeweiligen betroffenen Person" hat, gleichgültig ob Dritte oder Beschäftigte. [60] Die Aufzeichnung einer Videokonferenz darf also nicht durch beliebige Personen einsehbar sein, der Zugriff darauf muss auf den Teilnehmerkreis oder nur auf berechtigte Personen beschränkt sein.

Sollzustand Maßnahmen SDM D1.3	**Istzustand**
Implementierung eines sicheren Authentifizierungsverfahrens	• B: Authentifizierung über Linux-Sicherheitsmodell • B: Passwortkomplexität für Frontend festkodiert im Greenlight-Container (/usr/src/app/app/models/user.rb) • F: für Teilnehmer können lokale oder externe ID-Provider genutzt werden • F: für Räume können Zugriffscodes, für Moderatoren Zugangscodes erstellt werden • F: Zurücksetzen des Passwortes per E-Mail verschicktem Link
Festlegung und Kontrolle der Nutzung zugelassener Ressourcen insbesondere Kommunikationskanäle	• B: Dateiberechtigungen über Linux-Sicherheitsmodell nicht umgesetzt: Konfigurationen[37] inkl. Passwörter für DBs für Dritte lesbar, Inhaltsdateien für Dritte lesbar • B: Kommunikationsberechtigungen über Linux-Sicherheitsmodell (mitlesen der internen Kommunikation benötigt root-Rechte)

37 https://docs.bigbluebutton.org/administration/configuration-files/

Sollzustand Maßnahmen SDM D1.3	**Istzustand**
Verschlüsselung von gespeicherten oder transferierten Daten sowie Prozesse zur Verwaltung und zum Schutz der kryptografischen Informationen	nicht vorhanden
Schutz vor äußeren Einflüssen (Spionage, Hacking)	• B: Firewall aktiv • B: Spionage- / Hackangriff nicht ausgeführt

Tabelle 4 : Soll- und Istzustand für das Gewährleistungsziel Vertraulichkeit

Konkrete Maßnahmen wären:

- Bereitstellung einer Konfiguration der Passwortkomplexität
- Docker-Container im Rootless-Modus
- standardmäßige Verschlüsselung der internen Kommunikation
- verschlüsselte Speicherung der Inhaltsdateien
- Übermittlung der Logindaten als Hash
- Ende-zu-Ende-Verschlüsselung
- Implementierung einer Authentifizierung vor der Wiedergabe von Aufzeichnungen im Browser

6.4 Gewährleistungsziel Nichtverkettung

Die Nichtverkettung bezeichnet die Anforderung Daten, die für unterschiedliche Zwecke erhoben wurden, nicht zusammenzuführen (zu verketten oder zu verknüpfen). [61] Dies ist zugleich eine Anforderung der Zweckbindung des Art. 5 Nr. 1 b): personenbezogene Daten sind für „festgelegte, eindeutige und legitime Zwecke” zu erheben und „nicht in einer mit diesen Zwecken nicht zu vereinbarenden Weise” weiterzuverarbeiten. Das wäre z. B. der Fall, wenn die Daten der Meetingteilnehmer zur Leistungs- und Verhaltenskontrolle genutzt würden, also eine Verkettung von Arbeitszeitdaten und Meetingteilnahmedaten stattfände.

Sollzustand Maßnahmen SDM D1.4	**Istzustand**
Einschränkung von Verarbeitungs-, Nutzungs- und Übermittlungsrechten	• B: teilweise Einschränkung über Dateiberechtigungen des Linux-Sicherheitsmodell und Firewall → Leserechte für Dritte für Konfigurations- und Inhaltsdateien • B: in der Standardkonfiguration keine Einschränkung der Verarbeitungsrechte in den NoSQL-DBs • F: Zugriff auf Inhalte durch Konfiguration einschränkbar, in Standardkonfiguration deaktiviert
programmtechnische Unterlassung bzw. Schließung von Schnittstellen bei Verarbeitungsverfahren und Komponenten	Umsetzungsmöglichkeiten unbekannt
Trennung mittels Rollenkonzepten mit abgestuften Zugriffsrechten auf der Basis eines Identitätsmanagements	• teilweise betreiberabhängig • B: Nutzerberechtigungen über Linux-Sicherheitsmodell • F: Rechte und Rollen konfigurierbar • F: Identitätsmanagement durch ID-Provider möglich
Zulassung von nutzerkontrolliertem Identitätsmanagement	F: umgesetzt
Einsatz von zweckspezifischen Pseudonymen, Anonymisierungsdiensten, anonymen Credentials, Verarbeitung pseudonymer bzw. anonymisierter Daten	B: Pseudonymisierung durch Nutzung der „mapped User-ID”, aber z. T. mit zusätzlicher Verarbeitung des Namens aufgehoben (z. .B. bei Chatdaten oder in Logs)

Tabelle 5 : Soll- und Istzustand für das Gewährleistungsziel Nichtverkettung

Konkrete Maßnahmen wären:

- standardmäßig datenschutzgerechte Voreinstellungen im Frontend
- siehe Integrität und Vertraulichkeit

6.5 Gewährleistungsziel Transparenz

Die Transparenz fordert die jeweilig notwendige Nachvollziehbarkeit der Zwecke und Verarbeitung: Wer hat welche Daten wann und für welchen Zweck erhoben und verarbeitet? Diese Transparenz gilt es dem Betroffenen und den Aufsichtsbehörden im erforderlichen Maß als Nachweis- und Rechenschaftspflicht nachzukommen (Art. 5 Abs. 2 DSGVO). Für eine Videokonferenz bedeu-

tet dies z. B., dass das Stattfinden einer Aufzeichnung des Meetings den Teilnehmern bekanntgemacht wird und sie über den Zweck der Aufzeichnung in Kenntnis gesetzt werden.

Sollzustand Maßnahmen SDM D1.5	**Istzustand**
Dokumentation der Bestandteile von Verarbeitungstätigkeiten insbesondere der Geschäftsprozesse, Datenbestände, Datenflüsse und Netzpläne, dafür genutzte IT-Systeme, Betriebsabläufe, Beschreibungen von Verarbeitungstätigkeiten, Zusammenspiel mit anderen Verarbeitungstätigkeiten	Dokumentation Bestandteile, Datenflüsse und Netzpläne teilweise vorhanden
Dokumentation von Einwilligungen, deren Widerruf sowie Widersprüche	nicht vorhanden, Einbindung per API unbekannt
Protokollierung von Zugriffen und Änderungen	nicht vorhanden, Einbindung per API unbekannt
Versionierung	vorhanden
Dokumentation der Verarbeitungsprozesse mittels Protokollen auf der Basis eines Protokollierungs- und Auswertungskonzepts	nicht vorhanden, Einbindung per API unbekannt
Dokumentation der Quellen von Daten, bspw. des Umsetzens der Informationspflichten gegenüber Betroffenen, wo deren Daten erhoben wurden sowie des Umgangs mit Datenpannen	Informationen können im Frontend mit Link zu eigenen Datenschutzinformationen eingebunden werden
Berücksichtigung der Auskunftsrechte von Betroffenen im Protokollierungs- und Auswertungskonzept	Informationen können im Frontend mit Link zu eigenen Datenschutzinformationen eingebunden werden
Bereitstellung von Informationen über die Verarbeitung von personenbezogenen Daten an Betroffene	Informationen können im Frontend mit Link zu eigenen Datenschutzinformationen eingebunden werden

Tabelle 6 : Soll- und Istzustand für das Gewährleistungsziel Transparenz

Konkrete Maßnahmen wären:

- vollständige Dokumentation der Funktionsweise
- Bereitstellung der Möglichkeit zum Speichern von Einwilligungen, Widersprüchen u. ä. in der PostgreSQL-Datenbank
- Protokollierung von Anpassungen der Konfigurationen im Front- und Backend

6.6 Gewährleistungsziel Intervenierbarkeit

Intervenierbarkeit stellt die Anforderung, dass die Rechte der Betroffenen wahrgenommen werden können: Benachrichtigung, Berichtigung, Auskunft, Widerspruch, Datenübertragbarkeit, Einschränkung der Verarbeitung, Löschung, Eingriff in automatisierte Einzelentscheidungen. [62] Stellt ein Teilnehmer im Nachhinein fest, dass seine aufgezeichnete Bildschirmfreigabe etwas Vertrauliches offenbarte, so kann er die Löschung der Aufzeichnung einfordern.

Sollzustand Maßnahmen SDM D1.6	**Istzustand**
Maßnahmen für differenzierte Einwilligungs-, Rücknahme- sowie Widerspruchsmöglichkeiten	• nicht vorhanden • bei Meetingteilnahme muss Häkchen für evtl. Aufzeichnung gesetzt werden, sonst keine Teilnahme möglich
Schaffung notwendiger Datenfelder z. B. für Sperrkennzeichen, Benachrichtigungen, Einwilligungen, Widersprüche, Gegendarstellungen	nicht vorhanden
dokumentierte Bearbeitung von Störungen, Problembearbeitungen und Änderungen an Verarbeitungstätigkeiten sowie an den technischen und organisatorischen Maßnahmen	betreiberabhängig
Deaktivierungsmöglichkeit einzelner Funktionalitäten ohne Mitleidenschaft für das Gesamtsystem	F: Aufzeichnungen, Teilen von Räumen, Konferenzstart für alle Nutzer können deaktiviert werden B: unbekannt
Implementierung standardisierter Abfrage- und Dialogschnittstellen für Betroffene zur Geltendmachung und/oder Durchsetzung von Ansprüchen	nicht vorhanden, mögliche Implementierung unbekannt
Betreiben einer Schnittstelle für strukturierte, maschinenlesbare Daten zum Abruf durch Betroffene	unbekannt
operative Möglichkeit zur Zusammenstellung, konsistenten Berichtigung, Sperrung und Löschung aller zu einer Person gespeicherten Daten	teilweise manuell möglich F: Nutzerkonto durch Nutzer löschbar
Bereitstellen von Optionen für Betroffene, um Programme datenschutzgerecht einstellen zu können	nicht vorhanden (keine Nutzung von Chat, Notizen, Audio und Video möglich)

Tabelle 7 : Soll- und Istzustand für das Gewährleistungsziel Intervenierbarkeit

Konkrete Maßnahmen wären:

- Bereitstellung von Datenexports für einzelne Nutzer (z. B. im Frontend)
- Bereitstellung der Möglichkeit einer Meeting- / Konferenzteilnahme ohne Einwilligung zur Aufzeichnung inkl. automatisierter Deaktivierung der Aufzeichnung für den eingetretenen Fall
- Bereitstellung der Möglichkeit zum Speichern von Sperrvermerken in der PostgreSQL-Datenbank inkl. Sperrung der zugehörigen Inhaltsdateien

6.7 Gewährleistungsziel Datenminimierung

Datenminimierung kann mit dem Satz „So viele Daten wie nötig, so wenige Daten wie möglich.” beschrieben werden. Die Verarbeitung personenbezogener Daten soll auf das dem Zweck angemessene, erhebliche und notwendige Maß beschränkt werden (vgl. auch Erwägungsgrund 39 DSGVO). Das schließt eine Verarbeitung von Daten aus, die nicht für die Erreichung des Zwecks erforderlich sind. Dieser Grundsatz gilt nicht nur für den Umfang der Daten, sondern auch für die Verarbeitungsprozesse selbst. [57] Besteht der Verarbeitungszweck im Abhalten einer Videokonferenz, würde eine Verarbeitung des Geburtsdatums der Teilnehmer der Anforderung der Datenminimierung nicht entsprechen.

Sollzustand Maßnahmen SDM D1.7	**Istzustand**
Reduzierung erfasster Attribute der betroffenen Personen	F: Verarbeitung auf erforderliche Attribute beschränkt
Reduzierung der Verarbeitungsoptionen in Verarbeitungsprozessschritten	• B: redundante Verarbeitung von Attributen durch die Nutzung dreier DBMS (vgl. 5.2) • B: Log-Inhalte nicht auf ID-Attribute beschränkt (z. T. Teilnehmernamen in den Logs)
Reduzierung von Möglichkeiten der Kenntnisnahme vorhandener Daten	• F: Aufzeichnungen ohne Authentifizierung abspielbar - trotz Sichtbarkeit „Geschützt” • B: Verzeichnisse / Dateien mit Inhaltsdaten für Dritte lesbar • B: Daten in Redis und MongoDB können von normalen Nutzern abgefragt werden
Voreinstellungen für betroffene Personen	• F: virtueller Hintergrund einstellbar

Sollzustand Maßnahmen SDM D1.7	**Istzustand**
zur Beschränkung der Verarbeitung auf das erforderliche Maß	• F: keine Möglichkeiten für Löschung teilnehmerbezogener Webcam-, Chat- oder Notiz-Aufzeichnungen
Bevorzugung automatisierter Verarbeitungsprozesse zur Begrenzung einer Kenntnisnahme	B: durchgehend automatisiert
automatische Sperr- und Löschroutinen	• B: teilweise vorhanden durch Löschung bestimmter Logs bei Neustart • B: Cronjob zum Löschen der Inhaltsdaten nach n Tagen (inaktiv) • B: Sperr-Routine nicht vorhanden
Pseudonymisierungs- und Anonymisierungsverfahren	• B: Pseudonymisierung durch Nutzung der „mapped User-ID”, aber z. T. mit zusätzlicher Verarbeitung des Namens aufgehoben, wird nicht durchgängig genutzt: in PostgeSQL andere ID • B: IP-Adressen in Logs nicht pseudonymisiert • Bedarf für Anonymisierung unbekannt

Tabelle 8 : Soll- und Istzustand für das Gewährleistungsziel Datenminimierung

Konkrete Maßnahmen wären:

- Reduzierung der redundanten Datenverarbeitung und verwendeten Attribute (wenn User-ID vorhanden → Name nicht nötig)
- Reduzierung und Pseudonymisierung von Log-Inhalten (Änderung der Standardkonfiguration von DEBUG auf INFO)
- Bereitstellung von nutzerbezogenen Löschmöglichkeiten im Frontend
- Aktivierung des Cronjobs zum Löschen der Daten
- siehe Integrität und Vertraulichkeit

7 Fazit

Eventuelle durch den Betreiber oder Verantwortlichen umsetzbare Maßnahmen sind von dieser Einschätzung ausgenommen.

Der Istzustand in der Standardkonfiguration erfüllt weder die Anforderungen der Gewährleistungsziele noch die Grundsätze „Datenschutz durch Technik" (data protection by design) und „datenschutzfreundliche Voreinstellungen" (data protection by default)[38], sondern offenbart Mängel, die zur Schlussfolgerung des nicht datenschutzkonformen Einsatzes in der Standardkonfiguration führen.

Ein Einwilligungsmanagement ist nicht vorhanden, einer Aufzeichnung muss zugestimmt werden, ansonsten ist die Konferenz- oder Meetingteilnahme nicht möglich. Die Standardeinstellungen der „Raum-Konfiguration" im Frontend entsprechen nicht den datenschutzfreundlichen Voreinstellungen, da sie standardmäßig deaktiviert sind. Im Frontend gelöschte Aufzeichnungen bleiben im Dateisystem gespeichert.

Auf Betriebssystemebene der BBB-Installation wird das Linux-Sicherheitsmodell nicht konsequent umgesetzt. Das führt zu weitreichenden Berechtigungen und ermöglicht in der Konsequenz das Auslesen von Daten und Konfigurationen einschließlich Passwörtern durch nicht privilegierte Nutzer (u. U. unbefugte Personen). Administratoren müssen sich zahlreiche, z. T. nur bei Softwareherstellern verfügbare Informationen beschaffen, um Konfigurationen anpassen oder Funktionsweisen nachvollziehen zu können. Die Komponenten liegen z. T. in veralteten Versionen vor, was die Frage nach einer Update-Strategie aufwirft. Die Docker-Container nutzen auf dem Host das Verzeichnis eines privilegierten Nutzers statt der Verzeichnisse des Applikationsnutzers und sind nicht für den Rootless Modus konfiguriert. Der Ausbruch aus dem Container mit root-Rechten durch die Kompromittierung des Frontends wurde nicht getestet.

38 Erwägungsgrund 78 DSGVO „[…] Strategien festlegen und Maßnahmen ergreifen, die insbesondere den Grundsätzen des Datenschutzes durch Technik (data protection by design) und durch datenschutzfreundliche Voreinstellungen (data protection by default) Genüge tun."

Die drei Datenbanksysteme verarbeiten die Daten vielfach redundant. Das ist für eventuelle Ausfälle und Wiederherstellung von Prozess- und Datenzuständen hilfreich, allerdings führt es zu einer unnötigen Datenverarbeitung.

Es können nicht nur die Daten in den NoSQL-Datenbanken ohne Authentifizierung abgefragt und geändert werden. Auch das Abspielen von Aufzeichnungen ist ohne Authentifizierung möglich, obwohl die Sichtbarkeit der Aufzeichnung „Geschützt" ist.

Es werden nur die für das Abhalten eines Meetings tatsächlich notwendigen Daten erhoben und verarbeitet und diese soweit möglich durch MD5- oder SHA*-Routinen pseudonymisert, so dass eine Nachverfolgung oder übergreifende Verknüpfung von Nutzerinformationen zumindest erschwert wird. Allerdings ist sie keineswegs ausgeschlossen und mit Kenntnis der Datenstrukturen durchführbar. Durch die zusätzliche Speicherung von Namensattributen wird die Pseudonymisierung teilweise aufgehoben. Auch wird sie nicht konsequent umgesetzt, so dass Teilnehmeraktivitäten aus Log-Dateien nachvollziehbar sind. Das ist für eine Fehlersuche durchaus von Bedeutung, aber nicht im funktionierenden Betrieb.

Die Einbindung eigener Zertifikate, z. B. für die Nutzung einer PKI, gestaltet sich schwierig, weil die Installationsroutine zwar ein vorgegebenes Verzeichnis abfragt, aber die Zertifikate nicht den Komponenten bekanntmacht. Eine Ende-zu-Ende-Verschlüsselung z. B. durch Secure Frames (SFrame)[39], ist nicht vorgesehen. Das schließt BigBlueButton für die Verarbeitung von besonderen Kategorien personenbezogener Daten aus.

Vor- und Nachteil besteht in den umfangreichen Änderungsmöglichkeiten der Konfigurationen jeder genutzten Komponente. Auf der einen Seite ist die Gefahr, dass durch die Komplexität der Plattform datenschutzkonforme Einstellungen übersehen werden und es dadurch zu Datenschutzverletzungen kommt, nicht von der Hand zuweisen. Auf der anderen Seite bietet es die von Open-Source-Software gewohnte Anpassung an eigene Bedürfnisse.

39 https://www.rfc-editor.org/rfc/rfc9605.html

8 Generische Konfigurationshilfe

Einige Mängel sollten nach der Analyse des Istzustandes behoben werden. Angedacht war eine Anpassung der Standardeinstellungen für das Frontend, eine Korrektur der Berechtigungen im Dateisystem, die Aktivierung des Cronjobs zum Löschen der Daten und die Erstellung von Datenbank-Passwörtern. Da die BBB-eigenen Scripte ausschließlich Bash-Scripte sind, lag es nahe, eine automatisierte Korrektur ebenfalls auf diese Weise umzusetzen. BBB bietet die Möglichkeit in /etc/bigbluebutton von Neuinstallationen unveränderte Konfigurationen abzulegen[40]. Mit dem Script wurden die verbesserungswürdigen Parameter der Konfigurationen erfasst, korrigiert und die angepassten YAML- und JSON-Dateien im genannten Verzeichnis abgelegt. Für Redis wurde ein Passwort gesetzt. Das Resultat war ein nicht funktionierendes BBB. Die Fehlersuche ergab, dass ‚bbb-conf' die YAML-Dateien mithilfe von ‚yq'[41] zusammenführte, aber kommentierte Parameter den Absturz der WebRTC-Komponenten hervorriefen. Hier musste für den erfolgreichen Start die Originalkonfiguration geändert werden. JSON-Dateien, z. B. für Etherpad (Notizen), wurden von ‚bbb-conf' nicht geprüft und mussten ebenfalls im Original angepasst werden. Trotz der Übergabe einer Umgebungsvariable an den Greenlight-Container, wurde das Passwort nicht an das Backend weitergegeben. Hierfür musste ein Ruby-Script geändert werden. Für die Vergabe eines Passwortes der MongoDB hätte das Start-Script von Meteor verändert werden müssen. Die Änderungen der Zugriffsberechtigungen der Log-Verzeichnisse führten zur Fehlermeldung, dass diese nicht für alle lesbar seien. Insgesamt war eine Anpassung von 13 Konfigurationen und Scripten aller Komponenten notwendig. Diese Punkte führten schließlich zum Entschluss, keine automatisierte Konfigurationshilfe zur Verfügung zu stellen. Änderungen in Originaldateien, gleichgültig ob Konfiguration oder Script, sind grundsätzlich fehleranfällig, weil sie durch Aktualisierungen i. d. R. überschrieben werden. Wenn das Ziel eine funktionierende und datenschutzkonforme Installation ist, dann sollte eine Anpassung von Parametern über eine zentrale Konfiguration möglich sein. Eine Änderung der Konfigurationen jeder einzelnen Komponente ist nicht die Lösung.

40 https://docs.bigbluebutton.org/administration/configuration-files/

41 https://mikefarah.gitbook.io/yq

9 Zusammenfassung / Ausblick

Diese Arbeit deckt ein weites Spektrum der forensischen Untersuchung wie Prozess-, Datei-, Datenbank- und Netzwerkanalyse ab, geht jedoch auf wichtige Einzelaspekte wie Kryptographie, Arbeitsspeicherforensik oder Angriffsvektoren nicht ein. Vielmehr versucht sie einen Beitrag zum technischen Datenschutz zu leisten. Die eingangs gestellten Fragen konnten teilweise beantwortet werden.

- Es wurde aufgezeigt, welche personenbezogenen Daten auf welche Weise verarbeitet werden.
- Es wurden konkrete technische Maßnahmen genannt, die seitens des Softwareherstellers implementiert werden können.
- Die aus der forensischen Analyse gewonnenen Erkenntnisse konnten nicht in einer Konfigurationshilfe zur Verfügung gestellt werden. Mit einem Bash-Script war es möglich, die Anforderungen des Standard-Datenschutzmodells in technische Maßnahmen zu überführen, jedoch ist der Einsatz in keinster Weise für den Produktionsbetrieb zu empfehlen.

Die Arbeit zeigt, dass die Anwendung von Open-Source-Software nicht per se datenschutzkonform ist, aber für eigene Bedürfnisse angepasst werden kann und so die Möglichkeit bietet, das Datenschutzniveau anzuheben. Die Probleme sind lösbar, indem der Fokus zusätzlich zum reibungslosen Betrieb auch die Datenschutzbelange einbezieht. So wird beispielsweise bei Jitsi, das als Modul eingebunden werden kann, an der Implementierung einer Ende-zu-Ende-Verschlüsselung gearbeitet[42]. Mit der neuen BigBlueButton-Version wird vielleicht auch seine Komplexität reduziert. Keep it simple.

Wünschenswert ist ein konsistenter Maßnahmekatalog der Datenschutzaufsichtsbehörden, der eine tatsächliche Hilfe für Softwareentwickler ist und bei der technischen Umsetzung unterstützt. Mit der zunehmenden Digitalisierung wird die Zusammenarbeit von Recht und Technik auch für den Datenschutz essentieller. Das allerdings ist eine fast 50 Jahre alte Erkenntnis.

42 https://github.com/jitsi/lib-jitsi-meet/blob/master/doc/e2ee.md

Literaturverzeichnis

[1] Gilbert, Nestor : 71 Video and Web Conferencing Software Statistics: 2024 Analysis of Data & Market Share, FinancesOnline, 2024, https://financesonline.com/video-web-conferencing-software-statistics, Aufgerufen: 13.11.2024

[2] Suduc A.M.; Bizoi, M.; Filip, F.G.: Status, Challenges and Trends in Videoconferencing Platforms, In: International Journal of Computers Communications & Control, Online ISSN 1841-9844, ISSN-L 1841-9836, Volume 18, 2023, https://doi.org/10.15837/ijccc.2023.3.5465,

[3] Adavelli, Muninder : 19 Video Conferencing Statistics to Know in 2024, TechJury.net, 2024, https://techjury.net/video/video-conferencing-statistics/, Aufgerufen: 13.11.2024

[4] Vollmer, Anna Maria; Schmitt, Anna : Was ist Open Source Software? Definition, Vor- und Nachteile, Lizenzen, Blog des Fraunhofer-Institut für Experimentelles Software Engineering, 2020, https://www.iese.fraunhofer.de/blog/open-source-software/, Aufgerufen: 13.11.2024

[5] RedHat : Was ist Open Source?, RedHat, 2023, https://www.redhat.com/de/topics/open-source/what-is-open-source-software, Aufgerufen: 13.11.2024

[7] Cimpanu, Catalin : Hacker ransoms 23k MongoDB databases and threatens to contact GDPR authorities , ZDNET, 2020, https://www.zdnet.com/article/hacker-ransoms-23k-mongodb-databases-and-threatens-to-contact-gdpr-authorities/, Aufgerufen: 13.11.2024

[8] Melnikov, A. et al. : Simple Authentication and Security Layer (SASL), RFC 4422, IETF, Network Working Group, 2006, https://datatracker.ietf.org/doc/html/rfc4422, Aufgerufen: 13.11.2024

[9] Datanyze : Marktanteile der führenden Unternehmen für Video- und Audiokonferenzsysteme, Statista, 2024, https://de.statista.com/statistik/daten/studie/1228015/umfrage/marktanteile-der-fuehrenden-unternehmen-fuer-video-und-audiokonferenzsysteme/, Aufgerufen: 15.11.2024

[10] Termer Frank et al., Open Source Monitor 2023, bitkom, 2023, https://www.bitkom.org/opensourcemonitor2023

[11] Information Technology Laboratory; National Vulnerability Database : CVE-2021-45046 Detail, NIST, 2021, https://nvd.nist.gov/vuln/detail/CVE-2021-45046, Aufgerufen: 16.11.2024

[12] LfDI MV, 19. Tätigkeitsbericht zum Datenschutz, 3.6. Einsatz von Videokonferenzsystemen, S. 23 ff, Der Landesbeauftragte für Datenschutz und Informationsfreiheit Mecklenburg-Vorpommern, 2024

[13] DSK, Orientierungshilfe Videokonferenzsysteme, 2.1 Selbst betriebener Dienst, S. 5 f, Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder, 2020

[14] BlnDSB, Jahresbericht Datenschutz Berlin 2020, 1.4.1„Lernraum Berlin“, S. 50, Berliner Beauftragte für Datenschutz und Informationsfreiheit, 2021

[15] HBDI, 50. Tätigkeitsbericht Datenschutz, 4.2 Einsatz von Videokonferenzsystemen in Schulen und Hochschulen, S. 55 ff, Hessischer Beauftragter für Datenschutz und Informationsfreiheit, 2022

[16] BlnDSB, Hinweise für Berliner Verantwortliche zu Anbietern von Videokonferenzdiensten, Bewertung Teil 1, S. 4 ff, Berliner Beauftragte für Datenschutz und Informationsfreiheit, 2021

[17] BigBlueButton : Features, BigBlueButton Inc., 2024, https://bigbluebutton.org/features/, Aufgerufen: 17.11.2024

[18] LfDI Bremen, 5. Jahresbericht Datenschutz, 10.3 Videokonferenzsysteme im Schulkontext, S. 50 ff, Die Landesbeauftragte für Datenschutz und Informationsfreiheit der Freien Hansestadt Bremen, 2023

[19] TLfDI, 6. Tätigkeitsbericht zum Datenschutz nach der DS-GVO 2023, Vorwort, S.19 - 20, Thüringer Landesbeauftragter für den Datenschutz und die Informationsfreiheit, 2024

[20] BigBlueButton : Architecture, BigBlueButton Inc., 2024, https://docs.bigbluebutton.org/development/architecture, Aufgerufen: 17.11.2024

[21] BigBlueButton : API Reference, BigBlueButton Inc., 2024, https://docs.bigbluebutton.org/development/api/, Zugegriffen: 17. November 2024

[22] Gerling, Reiner W.; Gerling, Sebastian; Hessel, Stefan; Petrlic, Ronald, Stand der Technik bei Videokonferenzen - und die Interpretation der Aufsichtsbehörden, In: DuD Datenschutz und Datensicherheit, 11 / 2020, S. 740 - 747, 2020

[23] MongoDB : What is a Document Database?, MongoDB, Inc., 2024, https://www.mongodb.com/resources/basics/databases/document-databases, Aufgerufen: 20.11.2024

[24] MongoDB : What is NoSQL?, MongoDB, Inc., 2024, https://www.mongodb.com/resources/basics/databases/nosql-explained, Aufgerufen: 20.11.2024

[25] Kaufmann, Michael; Meier, Andreas : SQL- & NoSQL-Datenbanken, 9. Auflage, S. 16, Springer Vieweg, 2023. ISBN 978-3-662-67092-7

[26] Google for Developers : Erste Schritte mit WebRTC, Google, 2024, https://webrtc.org/getting-started/overview?hl=de, Aufgerufen: 20.11.2024

[27] Shekhada, Dhavalkumar; Stiller, Michael; Salvi, Aniket, A Comparison of Current Web Protocols for Usage in Cloud based Automation Systems, 2.2.1 WebSocket, S. 56, Kommunikation und Bildverarbeitung in der Automation, Technologien für die intelligente Automation, J. Jasperneite und V. Lohweg (Hrsg.), Springer-Verlag, 2018, https://doi.org/10.1007/978-3-662-55232-2_5

[29] BigBlueButton : Server Customization, Add a phone number to the conference bridge, BigBlueButton Inc., 2024, https://docs.bigbluebutton.org/administration/customize/#add-a-phone-number-to-the-conference-bridge, Aufgerufen: 21.11.2024

[30] Das Europäische Parlament und der Rat der Europäischen Union : Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates vom 27. April 2016, ABl. L 119 vom 4.5.2016, p. 1 - 88, Amtsblatt der Europäischen Union, 2016, https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=celex%3A32016R0679

[31] Europäischer Gerichtshof, Dritte Kammer : Urteil vom 9. November 2023, C-319/22, Rn. 46, Europäischer Gerichtshof, 2023, https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:62022CJ0319

[32] Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder (DSK), Orientierungshilfe Videokonferenzsysteme, Einleitung, S. 4, DSK, 2020

[33] Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder (DSK), Positionspapier zur biometrischen Analyse, 6.1.6.6 Gesichtsbilder, S. 21, DSK, 2019

[34] Der Landesbeauftragte für den Datenschutz Niedersachsen : Biometrie und Datenschutz, LfD Niedersachsen, 2024, https://www.lfd.niedersachsen.de/startseite/themen/technik_und_organisation/orientierungshilfen_und_handlungsempfehlungen/biometrie/biometrie-und-datenschutz-55984.html, Rn. 46: 22.11.2024

[35] Kagan, Dima; Fuhrmann Alpert, Galit; Fire, Michael, Zooming Into Video Conferencing Privacy and Security Threats, arXiv, 2020, https://doi.org/10.48550/arXiv.2007.01059

[36] Sun, Yuanyi; Zhu, Sencun; Chen, Yu, ZoomP3: Privacy-Preserving Publishing of Online Video Conference Recordings, PoPETs: Proceedings on Privacy Enhancing Technologies, Volume: 2022, Issue 3, S. 630 - 649, https://doi.org/10.56553/popets-2022-0089, 2022

[37] Robin1; Hermanto, Fransiskus; Chandra, Wenripin, Face Recognition Implementation as an Attendance Feature on Web-Based Video Conference Application, Brilliance: Research of Artificial Intelligence, Volume 3, Number 2, S. 282 - 289, https://doi.org/10.47709/brilliance.v3i2.3216, 2023

[38] BigBlueButton : Install Greenlight v3 , BigBlueButton Inc., 2024, https://docs.bigbluebutton.org/greenlight/v3/install/, Aufgerufen: 27.11.2024

[39] SUSE LLC and contributors : Introduction to systemd Basics, 1 What is systemd?, SUSE LLC, 2024, https://documentation.suse.com/smart/systems-management/html/systemd-basics/index.html, Aufgerufen: 15.11.2024

[40] Garrels, Machtelt : Introduction to Linux: Chapter 3. About files and the file system, The Linux Documentation Project (LDP), 2008, https://tldp.org/LDP/intro-linux/html/sect_03_01.html, Aufgerufen: 27.11.2024

[41] The IEEE and The Open Group : The Open Group Base Specifications Issue 8, IEEE Std 1003.1-2024, 3. Definitions, 3.141 File Descriptor, The IEEE and The Open Group, 2024, https://pubs.opengroup.org/onlinepubs/9799919799/basedefs/V1_chap03.html, Aufgerufen: 30.11.2024

[42] MongoDB : MongoDB Manual, WiredTiger Storage Engine, MongoDB, Inc., 2024, https://www.mongodb.com/docs/manual/core/wiredtiger/, Aufgerufen: 01.12.2024

[43] MongoDB : Reference Guide, WiredTiger command line utility, MongoDB, Inc., 2024, https://source.wiredtiger.com/11.3.1/command_line.html, Aufgerufen: 02.12.2024

[44] Kurogane, Akira : WiredTiger File Forensics Part 3: Viewing all the MongoDB Data, Percona LLC., 2021, https://www.percona.com/blog/wiredtiger-file-forensics-part-3-viewing-all-the-mongodb-data/, Aufgerufen: 02.11.2024

[45] MongoDB : MongoDB Manual, Storage, Journaling, MongoDB Inc., 2024, https://www.mongodb.com/docs/manual/core/journaling/, Aufgerufen: 02.12.2024

[46] Redis : Develop with Redis, Quick Starts, Redis FAQ, Redis, , https://redis.io/docs/latest/develop/get-started/faq/, Aufgerufen: 02.11.2024

[47] PostgreSQL : About, What is PostgreSQL?, The PostgreSQL Global Development Group, 2024, https://www.postgresql.org/about/, Aufgerufen: 02.12.2024

[48] Gentoo Authors : SELinux/Tutorials/The security context of a process, Privileges of processes, Gentoo Foundation, Inc., 2022, https://wiki.gentoo.org/wiki/SELinux/Tutorials/The_security_context_of_a_process#Confining_applications, Aufgerufen: 03.12.2024

[49] Morris, James : Overview of Linux Kernel Security Features, The Linux Foundation, 2013, https://www.linux.com/training-tutorials/overview-linux-kernel-security-features/, Aufgerufen: 03.12.2024

[50] Garrels, Machtelt : Introduction to Linux - A Hands on Guide, 3.4.1. Access rights: Linux's first line of defense, V 1.27, The Linux Documentation Project, 2008, https://tldp.org/LDP/intro-linux/html/intro-linux.html, Aufgerufen: 03.12.2024

[51] Redis : Develop with Redis, Redis reference, Redis serialization protocol specification , Redis, 2024, https://redis.io/docs/latest/develop/reference/protocol-spec/, Aufgerufen: 03.12.2024

[52] Däubler, Wolfgang; Wedde, Peter, Weichert, Thilo; Sommer, Imke : EU-DSGVO und BDSG Kompaktkommentar, 2. Auflage, DSGVO Art. 4 38 ff, BUND Verlag, 2020. ISBN 978-3-7663-6865-2

[55] Rost, Martin et al., Das Standard-Datenschutzmodell, Version 3.1, Impressum, S. 2 , AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder, 2024

[56] Rost, Martin et al., Das Standard-Datenschutzmodell, Version 3.1, Einleitung, S. 5 ff, AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder, 2024

[57] Rost, Martin et al., Das Standard-Datenschutzmodell, Version 3.1, C1.1 Datenminimierung (Dm), S. 25, AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder, 2024

[58] Rost, Marin et al., Das Standard-Datenschutzmodell, Version 3.1, C1.2 Verfügbarkeit (Vf), S. 25 f, AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder,

[59] Rost, Martin et al., Das Standard-Datenschutzmodell, Version 3.1, C1.3 Integrität (Ig), S. 26, AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder, 2024

[60] Rost, Martin et al., Das Standard-Datenschutzmodell, Version 3.1, C1.4 Vertraulichkeit (Vt), S. 26, AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder, 2024

[61] Rost, Marin et al., Das Standard-Datenschutzmodell, Version 3.1, C1.5 Nichtverkettung (Nn), S. 27, AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder, 2024

[62] Rost, Martin et al., Das Standard-Datenschutzmodell, Version 3.1, C1.7 Intervenierbarkeit (Iv), S. 27 f, AK Technik der Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder,

Bilderverzeichnis

Tabellenverzeichnis

Anlage Anpassung der VM für den BigBlueButton-Betrieb

Abschalten von IPv6

```
# grep ipv6 /etc/sysctl.conf | grep -v ^#
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Anpassung der FreeSWITCH-Konfiguration:

```
# grep "listen-ip"
  /opt/freeswitch/etc/freeswitch/autoload_configs/event_socket.conf.xml
    <param name="listen-ip" value="127.0.0.1"/>
```

E-Mail-Versand

Abweichungen von der Postfix-Standardkonfiguration:

```
# grep sasl /etc/postfix/main.cf
smtp_sasl_auth_enable = yes
smtpd_sasl_auth_enable = yes
  permit_sasl_authenticated,

# cat /etc/postfix/sasl/smtpd.conf
pwcheck_method: saslauthd
mech_list: PLAIN LOGIN
saslauthd_path: /var/run/saslauthd/mux
```

Konfiguration SASL:

```
# grep -v ^# /etc/default/saslauthd
  START=yes
DESC="SASL Authentication Daemon"
NAME="saslauthd"
MECHANISMS="shadow"
MECH_OPTIONS=""
THREADS=5
OPTIONS="-c -m /var/run/saslauthd"
```

SMTP-Parameter für Docker-Container „greenlight-v3” :

```
# grep SMTP ~/greenlight-v3/.env | grep -v ^#
SMTP_SERVER=blue.domain.de
SMTP_PORT=25
SMTP_USERNAME=blue@domain.de
SMTP_PASSWORD=G4nzFurchtb4rG3h31m.Wirklich!
SMTP_AUTH=plain
```

```
SMTP_DOMAIN=domain.de
SMTP_SENDER_EMAIL=blue@domain.de
SMTP_SENDER_NAME=blue
SMTP_STARTTLS=false
SMTP_TLS=false
```

Firewallfreischaltung:

```
# ufw allow from 172.18.0.0/24 to 192.168.1.124 port 25
```

Installation selbsterstellter Zertifikate

Erstellung Zertifikate

```
# openssl genrsa -aes256 -out ca_key.pem 4096
# openssl req -x509 -new -nodes -extensions v3_ca -key ca_key.pem -days 365 -out
  ca_cert.pem -sha512
# openssl genrsa -out blue_key.pem 4096
# openssl req -new -key blue_key -out blue.csr   -sha512
# openssl x509 -req -extfile <(printf
  "subjectAltName=DNS.1:tuxbook.de,DNS.2:blue.tuxbook.de,DNS.3:www.tuxbook.de") -
  in blue.csr -CA ca_cert.pem -CAkey ca_key.pem -CAcreateserial -out blue.pem -
  days 365 -sha512
```

Kopieren der Zertifikate in das während der Installation abgefragte Verzeichnis:

```
# cat ~/blue.pem ~/ca_cert.pem > /local/certs/fullchain.pem
# cp ~/ca_key.pem /local/certs/privkey.pem
# cp ~/ca_cert.pem /local/certs/
# openssl x509 -in ~/ca_cert.pem -outform DER -out ~/ca_cert.crt
```

Aktualisierung des CA-Trust-Bundles:

```
# cp ~/ca_cert.crt /usr/local/share/ca-certificates/
# update-ca-certificates
```

Implementierung im Docker-Image:

```
# mkdir ~/greenlight-v3/ssl; cp ~/ca_cert.crt ~/greenlight-v3/ssl/
# cat ~/greenlight-v3/Dockerfile
FROM bigbluebutton/greenlight:v3
COPY ./ssl/ca_cert.crt /root/

RUN cp /root/ca_cert.crt /usr/local/share/ca-certificates \
       && /usr/sbin/update-ca-certificates --fresh
```

Anpassung der Compose-Definition:

```
# cat ~/greenlight-v3/docker-compose.yml
...
  greenlight-v3:
    build:
      context: .
      dockerfile: Dockerfile
    entrypoint: [bin/start]
...
```

Neuerzeugung des Docker-Images:

```
# cd ~/greenlight-v3
# docker-compose -f docker-compose.yml up -d
```

Bekanntmachung für Meteor-Komponente:

```
# grep ca_cert /usr/share/meteor/bundle/systemd_start.sh
export NODE_EXTRA_CA_CERTS=/local/certs/ca_cert.pem
```

Hinzufügen zum Java-Keystore:

```
# keytool -import -alias CA -keystore /etc/ssl/certs/java/cacerts -file
  /local/certs/ca_cert.pem
```

Verzeichnis der Abkürzungen

AOF	Append Only File
API	Application Programming Interface
BBB	BigBlueButton
BSON	JSON im Binärformat
CLI	Command Line Interface
DB	Datenbank
DBMS	Datenbanksystem
DDP	Distributed Data Protocol
E2EE	End-to-End-Encryption
FIFO	first-in-first-out
HTML5	Hypertext Markup Language Version 5
HTTP	Hypertext Transfer Protocol
ID	Identifier, Identifikationsnummer
IPv4	Internetprotokoll V4
IPv6	Internetprotokoll V6
JSON	Javascript Object Notation
KI	Künstliche Intelligenz
LDAP	Lightweight Directory Access Protocol
LTI	Learning Tools Interoperability
MCU	Multipoint Control Units
MITM	Man-In-The-Middle
Oauth	Open Authorization
OIDC	OpenID Connect
OS	Operating System
OSS	Open-Source-Software
PID	Prozess Identifier
PKI	Public Key Infrastruktur
RAW	roh, Rohdaten
RDB	Redis Database
RESP	Redis Serialization Protocol
RTP	Real-Time Transport Protocol
SAML	Security Assertion Markup Language
SASL	Simple Authentication and Security Layer
SDM	Standarddatenschutzmodell
SFU	Selective Forwarding Unit
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VM	Virtual Machine
WebRTC	Web Real-Time Communication
XML	Extensible Markup Language

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Die Stellen der Arbeit, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Ich erkläre ferner, dass ich die vorliegende Arbeit in keinem anderen Prüfungsverfahren als Prüfungsarbeit eingereicht habe oder einreichen werde.

Die eingereichte schriftliche Arbeit entspricht der elektronischen Fassung. Ich stimme zu, dass eine elektronische Kopie gefertigt und gespeichert werden darf, um eine Überprüfung mittels Anti-Plagiatssoftware zu ermöglichen.

Ort, Datum	Unterschrift